

Pirouette Multivariate Data Analysis Software

Version 5.0

Infometrix, Inc.

Pirouette® is licensed to you by Infometrix, Inc. You may only use this software in accordance with the license agreement. You may use the software only on a single computer. You may not copy the software except for archive or backup purposes and you may not reverse engineer, decompile, or disassemble the software that comprises the Pirouette system.

Limited Warranty: Infometrix warrants that the software will perform substantially in accordance with the accompanying electronic documentation and optional written materials for a period of 90 days from the date of receipt. Infometrix further warrants that any hardware accompanying the software will be free from defects in materials and workmanship under normal use and service for a period of one year from the date of receipt. Any implied warranties on the software and hardware are limited to 90 days and one year, respectively. Some states do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

Customer Remedies: Infometrix' entire liability and your exclusive remedy shall be, at Infometrix' option, either (a) return of the price paid or (b) repair or replacement of the software or hardware that does not meet Infometrix' Limited Warranty and which is returned to Infometrix with proof of purchase. This Limited Warranty is void if failure of the software or hardware has resulted from accident, abuse or misapplication. Any replacement software will be warranted for the remainder of the original warranty period or 30 days, whichever is longer.

No Other Warranties: Infometrix disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the software, any accompanying hardware, and the accompanying written materials. This limited warranty gives you specific legal rights, but you may have others, which will vary from state to state.

No Liability for Consequential Damages: In no event shall Infometrix or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use this Infometrix product, even if Infometrix has been advised of the possibility of such damages. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

Infometrix, Pirouette and Ai-Metrix are registered trademarks
and InStep and LineUp are trademarks of Infometrix, Inc.
Other company names and product names are trademarks of their respective owners.

Copyright © 1985–2023 by Infometrix, Inc. All rights reserved.
Infometrix, Inc., 11807 North Creek Parkway S, Suite B-111, Bothell, WA 98011
Phone: (425) 402-1450

Information: info@infometrix.com, Support: support@infometrix.com
World Wide Web: <http://www.infometrix.com/>

Preface

Although the growth of computing technology has enabled users to collect ever increasing amounts of data, software in many respects, has not kept pace with how we use our hardware. In a day an analytical or process instrument can collect data on many samples, each with hundreds or thousands of variables. The software bundled with most instruments is not designed to extract meaningful information efficiently from such large data sets. Instead, the emphasis is on spewing (compiling and printing tables) and storing (archiving them for later retrieval). Moreover, although the data may be multivariate, most data analysis software treats it as a succession of non-correlated, univariate measures.

Today's technology demands a better approach: one that acknowledges not only the non-specific and multivariate nature of most instrumented data but also common bottlenecks in the data analysis process:

- a plethora of algorithms which can distract or even confuse the user
- the lack of a standard file format to ease the blending of data from several instrument sources
- non-intuitive and non-graphical software interfaces which steepen an already challenging learning curve
- the absence of a mechanism to organize *all* computations performed on a data set into a single file

Welcome to Pirouette

Pirouette was developed to address all of the problems mentioned above while also taking advantage of the stability and standardization of the current 32-bit Windows operating systems which permit virtually unlimited file size and true multitasking/multithreading (Pirouette will run as a 32-bit application on 64-bit systems).

One key strength of Pirouette is the complete integration of graphics. Typically, the result of an analysis is a graphic or group of graphics. These range from 2D plots and line plots to dendrograms and rotatable 3D plots. Multiple plots in different windows are automatically linked, where appropriate, so that when two or more graphics are on screen, samples highlighted in one display are also highlighted in the others. For example, samples highlighted in a principal component scores plot will also be highlighted in a dendrogram.

Performing analyses rapidly and easily is important; however, the saving and re-using the results of these analyses as models is equally important. With Pirouette, working with

model files is as easy as working with any other file. Any model created within Pirouette can be saved and re-loaded later; predictions on new samples do not require rebuilding the model.

We have audited the way the majority of our users work and found that a stand-alone manual, regardless of its quality, is consulted infrequently. We decided, therefore, to supply the documentation as an electronic help file. Complete information is at your electronic fingertip in the form of a portable document format, complete with hyperlinked text. To obtain a hardcopy of any portion of this user guide, simply print from the Acrobat Reader software.

We believe Pirouette to be the most powerful and yet easy to use statistical processing and display program available. The routines included in this version are broadly applicable and often encountered in the chemometric literature. The graphical representations of the data and the interactive windowing environment are unique. As we continue to refine the software interface and enhance the statistical features of Pirouette, we look forward to your comments.

Happy computing and thanks for selecting Pirouette!

Structure of the Documentation

The document you are reading is organized with the goal of training you in multivariate analysis, regardless of your chemometrics background or level of experience in windowing interfaces. The basic thrust of each major section is listed below. Several chapters refer to data sets included with Pirouette. If you follow along with our examples, you will better understand both the points made in the chapter and how to work with Pirouette to analyze your own data sets.

PART I INTRODUCTION TO PIROUETTE

This section briefly introduces the Pirouette environment, discusses the software installation, and explains how to run a Pirouette analysis and build both classification and regression models for future use.

Chapter 1, Quick Start This introductory chapter contains everything you need to get started with Pirouette. Basic features of the Pirouette environment are described, including data input, running algorithms and viewing data and results.

Chapter 2, Pattern Recognition Tutorial This chapter walks through the analysis of a classification data set to introduce the Pirouette environment and explain some of the thought processes behind multivariate analysis.

Chapter 3, Regression Tutorial This chapter walks through a detailed analysis of a regression data set to introduce the Pirouette environment and multivariate analysis. It can augment or replace the instruction given in Chapter 2.

PART II GUIDE TO MULTIVARIATE ANALYSIS

Part II explains how to perform a multivariate analysis with Pirouette while also serving as a textbook on the multivariate methods themselves.

Chapter 4, Preparing for Analysis This chapter discusses how to prepare data for analysis. Details of transforms and preprocessing options are included.

Chapter 5, Exploratory Analysis This chapter explains how to run an exploratory data analysis. The two exploratory algorithms contained in Pirouette, Hierarchical Cluster Analysis (HCA) and Principal Component Analysis (PCA), are explained in detail, along with a discussion of how to manipulate and interpret their graphical results.

Chapter 6, Classification Methods This chapter explains how to build a classification model and use it to classify unknown samples. Pirouette's two classification algorithms—K-Nearest Neighbor (KNN) and Soft Independent Modeling of Class Analogy (SIMCA)—are discussed in detail with an emphasis on how to interpret the results of each.

Chapter 7, Regression Methods This chapter explains how to build a multivariate regression model and use it to predict continuous properties for unknown samples. Pirouette's two factor-based regression algorithms—Partial Least Squares (PLS) and Principal Component Regression (PCR)—are discussed jointly in detail. The results of the two algorithms are interpreted separately and compared. Classical Least Squares (CLS) is also described and contrasted with PLS and PCR.

Chapter 8, Mixture Analysis This chapter describes methods used to resolve mixtures into their underlying components. Multivariate Curve Resolution can be used to deconvolve fused chromatographic peaks and can apportion mixtures into their source compositions.

Chapter 9, Examples This chapter contains a series of application vignettes which can be a starting point for your own specific work. We have built this chapter as an overview of different data sets; many are supplied with Pirouette so that you can experiment with the data yourself.

PART III SOFTWARE REFERENCE

Part III is a guide to the Pirouette graphical interface, giving helpful hints so that you can exploit its full power. This section also serves as a technical reference for the various buttons, menu options and features unique to the Pirouette environment.

Chapter 10, The Pirouette Interface This chapter explains how Pirouette's tools, cursors and buttons are used to manipulate and interact with tabular and graphical displays. In addition, we discuss linking of results shown in different screen windows and how to create data subsets from the graphic display.

Chapter 11, Object Management This chapter explains how to use a central component of Pirouette, the Object Manager, for accessing data subsets and computed results.

Chapter 12, Charts This chapter describes the various types of graphs available in Pirouette, along with explanations of how to navigate and manipulate each.

Chapter 13, Tables This chapter describes how to hand enter and modify data within the spreadsheet. Also included are explanations of how to create subsets from tables and the navigation, sorting and editing tools.

Chapter 14, Data Input This chapter discusses reading and merging existing files to form a project-oriented Pirouette file.

Chapter 15, Output of Results This chapter explains how to print and save Pirouette objects and files, including how to save Pirouette model files for use in future predictions.

Chapter 16, Pirouette Reference This chapter describes all Pirouette menu options and dialogs.

PART IV APPENDICES

A series of subjects are addressed in appendices, including troubleshooting suggestions.

Chapter 17, An Introduction to Matrix Math This chapter gives a background in the matrix mathematics underlying all of Pirouette's multivariate algorithms. In addition, it describes nomenclature used in presenting equations.

Chapter 18, Tips and Troubleshooting This chapter details the error messages in Pirouette and tips on what you may be able to do when confronted with an error.

Chapter 19, Pirouette Scripting This chapter discusses the use and gives examples for processing data using Pirouette engine, but without invoking the user interface. Rules and commands are outlined along with some examples.

PIROUETTE RELEASE NOTES

A "Release Notes" document accompanies the materials that comprise Pirouette. Peruse this file to learn about new features and enhancements in recent versions as well as known problems.

ACKNOWLEDGMENTS

The creation of a product like Pirouette involves a series of trade-offs pitting completeness against ease of use. Through this tug-of-war, we have pushed for a comprehensive approach from an application standpoint, not concerning ourselves with trying to span every variant on a theme. Pirouette development has always been a team effort which traces its origins back to the late 1970s and early 1980s when the desire for such a product was strong, but the tools to create it were weak. The team included a number of special individuals in government, academia and industry who supplied comments and provided down-to-earth applications. We particularly wish to acknowledge the suggestions and input of the following individuals:

Fred Fry	Rick Murray
David Haaland	Randy Pell
Arnd Heiden	Mary Beth Seasholtz
Russell Kaufman	Takehiro Yamaji
Vanessa Kinton	

with more recent contributions from:

Greg Banik	Rick Murray
Fred Fry	Andrea Rauch
Peter Gibson	

It is also appropriate to mention the employees who contributed significantly to prior versions of Pirouette and related Infometrix products. The efforts of Barry Neuhaus, Susannah Bloch, Gerald Ring-Erickson, Meeji Ko, Carol Li, Stacey Miller, David Summers, Tom Wanne, Dean Webster and Joseph Yacker all helped lay the groundwork for this software.

The team most responsible for the current version of Pirouette is comprised of the following phenomenal individuals: Scott Ramos, Jay Matchett, Marlana Blackburn, Meeji Ko, Randy Pell, and Brian Rohrback. Without their dedication, the Pirouette product would not have attained its current form.

We would like to acknowledge the following individuals who contributed to the translations of the user interface.

French:

Jean-François Antinelli	Analytics Consulting, Nice, France;
Regis Grenier	Bio-Rad Laboratories

German:

Arnd Heiden and Carlos Gil	Gerstel GmbH & Col. KG, Mulheim an der Ruhr, Germany
-------------------------------	--

Italian:

Giuliana Drava	Dept of Pharmaceutical and Food Chemistry, University of Genova, Italy
----------------	--

Japanese:

Masato Nakai and Takehiro Yamaji	GL Sciences, Tokyo, Japan
-------------------------------------	------------------------------

Portuguese:

Scott Ramos	Infometrix, Bothell, WA
-------------	-------------------------

Spanish:

Scott Ramos, Vanessa Kinton and Rodolfo Romañach	Infometrix, Bothell, WA; Washington, DC; and Universidad de Puerto Rico, Mayagüez, Puerto Rico
--	--

This user guide was produced using Adobe's FrameMaker publishing package, and converted to a portable document format for online viewing.

Contents

Part I. Introduction to Pirouette

Chapter 1 Quick Start

Pirouette Briefly	1-1
Data Input	1-2
Running Algorithms	1-2
Viewing Results	1-3
The Object Manager	1-4
Saving Images	1-5
Saving Data and Models	1-5
Pirouette Help	1-6
Technical Support	1-6

Chapter 2 Pattern Recognition Tutorial

The Basics	2-1
Define the Problem	2-1
Open the File	2-2
Examine the Data	2-4
Exploratory Analysis	2-7
Running Exploratory Algorithms	2-7
Data Interpretation	2-10
Modeling and Model Validation	2-18
KNN Modeling	2-18
Review	2-23
References	2-23

Chapter 3 Regression Tutorial

The Basics	3-1
Define the Problem	3-1
Organize the Data	3-2
Read the File	3-2
Examine the Data	3-5
Exploratory Data Analysis	3-7
Set Up	3-7

Running the Exploration Algorithms	3-9
Data Interpretation	3-11
The Next Step: Calibration	3-16
Calibration and Model Validation	3-16
Set Up	3-17
Calibration with PCR and PLS	3-17
Data Interpretation	3-18
Saving the Model	3-27
Prediction of Unknowns	3-29
Review	3-33
References	3-33

Part II. Guide to Multivariate Analysis

Chapter 4 Preparing for Analysis

Overview	4-1
Defining the Problem	4-3
Organizing the Data	4-4
Assembling the Pieces	4-4
Training Set Structure	4-5
Checking Data Validity	4-5
Visualizing the Data	4-6
Line plots	4-6
Scatter Plots	4-8
Transforms	4-10
Viewing Transformed Data	4-10
Configuring Transforms	4-10
Preprocessing	4-26
Mean-Center	4-26
Variance Scale	4-27
Autoscale	4-29
Range Scale	4-30
Pareto Scale	4-31
Setting Preprocessing Options	4-32
Preprocessing and Outliers	4-33
Calibration Transfer	4-33
Subset selection	4-34
Additive and Multiplicative Adjustment	4-34
Direct and Piecewise Adjustment	4-34
Final Remarks	4-35
References	4-36

Chapter 5 Exploratory Analysis

Hierarchical Cluster Analysis	5-1
---	-----

	Mathematical Background	5-2
	HCA Objects	5-4
	Linkage Methods Illustrated	5-5
	Choosing a Linkage Method	5-10
	Principal Component Analysis	5-13
	General Concepts	5-13
	Mathematical Background	5-16
	Running PCA	5-31
	Making a PCA Prediction	5-43
	References	5-46
Chapter 6	Classification Methods	
	K Nearest Neighbors	6-2
	Mathematical Background	6-3
	Nearest Neighbor Example	6-3
	Running KNN	6-5
	Optimizing the Model	6-10
	HCA as a KNN Viewing Tool	6-12
	Making a KNN Prediction	6-12
	Soft Independent Modeling of Class Analogy	6-15
	Mathematical Background	6-16
	Running SIMCA	6-19
	Optimizing the Model	6-25
	Making a SIMCA Prediction	6-26
	Calibration Transfer	6-29
	Required class variables	6-29
	Calibration Transfer Options	6-29
	X Transferred	6-30
	References	6-30
Chapter 7	Regression Methods	
	Factor Based Regression	7-2
	Mathematical Background	7-3
	Orthogonal Signal Correction	7-12
	Running PCR/PLS	7-14
	Making a PCR/PLS Prediction	7-31
	PLS for Classification	7-38
	Running PLS-DA	7-38
	Making a PLS-DA Prediction	7-40
	Classical Least Squares	7-44
	Mathematical Background	7-44
	Running CLS	7-48
	Making a CLS Prediction	7-53
	Calibration Transfer	7-56
	Required variables	7-56
	calibration Transfer Options	7-57
	X Transferred	7-57
	Locally Weighted Regression	7-58

LWR Modeling	7-58
LWR Prediction	7-58
References	7-59

Chapter 8 Mixture Analysis

Introduction	8-1
Alternating Least Squares	8-3
Mathematical Background	8-3
Running ALS	8-4
Making an ALS Prediction	8-10
Multivariate Curve Resolution	8-12
Mathematical Background	8-15
Running MCR	8-19
Making a MCR Prediction	8-25
Reference	8-27

Chapter 9 Examples

Description of Example Files	9-1
Data Set References	9-5
Food and Beverage Applications	9-5
The Chemometric Approach	9-6
Specific Applications	9-6
Summary	9-9
Food and Beverage References	9-9
Environmental Science Applications	9-11
Specific Applications	9-12
Summary	9-15
Selected Environmental References	9-15
Chemometrics in Chromatography	9-17
Specific Applications	9-19
Summary	9-21
Selected Chromatography References	9-21

Part III. Software Reference

Chapter 10 The Pirouette Interface

Overview	10-1
Selecting in Lists and Tables	10-1
Selecting in Graphics	10-2
The Pirouette Window	10-2
Ribbon Buttons	10-3
File and processing functions	10-3
Window manipulations	10-4
Interaction Tools	10-4
Editing	10-4

View Switching	10-4
Plot Customization	10-5
Navigation Aids	10-5
Spinner Control	10-5
Cursors	10-6
View Preferences	10-7
Color Attributes	10-7
Text Attributes	10-8
Grid	10-9
Other Attributes	10-9
Chart Preferences	10-16
Label Attributes	10-16
Window Attributes	10-17
Color Sequence	10-18
Other Preferences	10-19
Prediction	10-19
Info Box Font	10-20
Sticky Features and Default Settings	10-20
Preference Sets	10-21
Language	10-21

Chapter 11 Object Management

The Object Manager Window	11-1
Navigation	11-2
Naming Conventions	11-3
Finding Objects	11-5
Renaming Objects	11-6
Deleting Objects	11-7
Charts	11-7
Creating Charts	11-8
Custom Charts	11-8
Subsets	11-9
Sample Selection	11-10
Variable Selection	11-11
References	11-12

Chapter 12 Charts

Creating Charts	12-1
Creating Charts from the Object Manager	12-1
Creating Charts with the Drop Button	12-3
Window Titles	12-3
Pirouette Graph Types	12-4
Scatter Plots	12-5
Specifying Axes	12-5
Selecting Points	12-5
Identifying Points	12-7
Point Labels	12-7

Cloaking	12-8
Magnifying Regions	12-8
Spinning a 3D Plot	12-9
Plot Scaling	12-11
Line Plots	12-13
Specifying Axes and Orientation	12-14
Identifying Lines	12-15
Magnifying Regions	12-15
Panning Line Plots	12-16
Axis Labels	12-16
Selecting Lines	12-17
Selecting Ranges	12-18
Redrawing Traces	12-19
Factor Selection Line Plots	12-19
Multiplots	12-20
The Dendrogram	12-22
The Dendrogram Environment	12-22
Dendrogram Navigation	12-25
Setting Similarity Values	12-25
Creating Class Variables	12-27
Identifying Samples	12-28
Linking Views	12-28
Creating Subsets from a Graphic	12-32
Plot Colors	12-34

Chapter 13 **Tables**

Introduction to the Spreadsheet	13-1
Navigating the Spreadsheet	13-2
Moving the Active Cell	13-2
Moving to a New Page	13-4
Selecting Data	13-5
Editing Data	13-7
Changing Data Values	13-8
Manipulating Ranges of Data	13-8
Changing Variable Types	13-10
Sorting Data	13-11
Transpose	13-12
Finding Missing Values	13-13
Filling Missing Values	13-13
Class Variables	13-19
Activating a Class Variable	13-19
Using Class Variables in Algorithms	13-19
Creating Subsets from Tables	13-20
Excluding Data	13-20
Including Data	13-21
Modifying Subsets	13-21
Sample and Variable Selection	13-22

	References	13-22
Chapter 14	Data Input	
	Entering New Data	14-1
	Opening and Merging Existing Data Files	14-3
	Common File Formats	14-5
	ASCII Files	14-5
	Excel Files	14-9
	Other File Formats	14-10
Chapter 15	Output of Results	
	Printing	15-1
	Capturing Chart Windows	15-2
	Saving Files	15-3
	Saving Data	15-4
	Saving Results	15-5
	Saving Models	15-6
	Pirouette Models	15-6
	ASCII Models	15-9
	Galactic Models	15-13
	References	15-13
Chapter 16	Pirouette Reference	
	Menu Features and Shortcuts	16-1
	File Menu	16-3
	New	16-3
	Open Data	16-4
	Save Data	16-5
	Save Data As	16-5
	Merge Samples, Merge Variables	16-5
	Save Object(s)	16-6
	Transpose	16-7
	Open Model	16-7
	Save Model	16-8
	Print	16-9
	Print Setup	16-10
	Recent Files	16-11
	Exit	16-11
	Edit Menu	16-11
	Undo	16-13
	Cut	16-13
	Copy	16-13
	Paste	16-14
	Clear	16-14
	Insert	16-14
	Delete	16-14
	Activate Class	16-15

No Class	16-15
Create Exclude/Exclude	16-15
Create Include/Include	16-16
Find Missing Values	16-16
Go To	16-17
Column Type.....	16-17
Sort	16-18
Fill.....	16-18
New Set.....	16-19
Process Menu.....	16-19
Run	16-20
Predict	16-31
Select Samples	16-32
Select Variables.....	16-33
Display Menu.....	16-34
Point Labels.....	16-35
Axis Labels	16-35
Plot Scaling	16-35
Zoom Current Plot	16-36
Unzoom Current Plot	16-36
Tools	16-36
Views	16-37
Selector.....	16-37
Cloak.....	16-37
Redraw	16-37
Limits	16-37
Labels	16-37
Objects Menu.....	16-38
Find	16-38
Rename	16-39
Expand Tree/Contract Tree.....	16-39
Create Chart	16-39
Windows Menu	16-39
Preferences	16-40
Cascade/Tile	16-44
Close Window/Close All Windows	16-44
Help Menu	16-45
Contents	16-45
Index	16-45
Release Notes	16-45
Setup	16-45
About Pirouette.....	16-46

Part IV. Appendices

Chapter 17	An Introduction to Matrix Math	
	Vectors and Matrices	17-1
	Matrix Operations	17-3
	Matrix Inversion	17-5
	Eigenvectors and Eigenvalues	17-6
	Reference	17-7
Chapter 18	Tips and Troubleshooting	
	Tips	18-1
	Frequently Asked Questions	18-2
	Messages	18-3
	Error Messages	18-3
	Warning Messages	18-8
	Other Alerts	18-12
	Known Problems	18-12
	Technical Assistance	18-14
Chapter 19	Pirouette Scripting	
	Introduction	19-1
	Rules and Instructions	19-2
	Scripting Commands - Alphabetical Order	19-3
	Scripting Commands - Functional Order	19-6
	Example Scripts	19-8

Part I.

Introduction to Pirouette

- 1 Quick Start**
- 2 Pattern Recognition Tutorial**
- 3 Regression Tutorial**

Quick Start

Contents

Pirouette Briefly	1-1
Technical Support	1-6

Welcome to Pirouette, part of the Infometrix family of easy-to-use multivariate analysis packages. This chapter is designed to get you up to speed in using Pirouette without referring to our extensive documentation. When you have browsed this chapter, you may want to follow one or both of the tutorials, in [Chapter 2, Pattern Recognition Tutorial](#) and [Chapter 3, Regression Tutorial](#).

Pirouette can be run in both demonstration and normal mode. The demonstration mode offers full functionality of the data processing and viewing components of the software, but analyzes only the example data files bundled with the package. It can, however, still be used to visualize any data set that can be loaded as well as to convert files in supported formats. Even Pirouette binary files (with a .PIR extension) can be opened and investigated: previously computed results can be viewed and evaluated. Thus, the demonstration version is itself a powerful data visualization package.

Users purchasing Pirouette will be issued a license which enables access to all chemometric algorithms available in the product. Instructions for licensing Pirouette, as well as any other Infometrix product, are detailed in a separate document—the [Licensing Guide for Infometrix Software](#)—that is installed with IPAK, the Infometrix Product Access Kit.

Pirouette Briefly

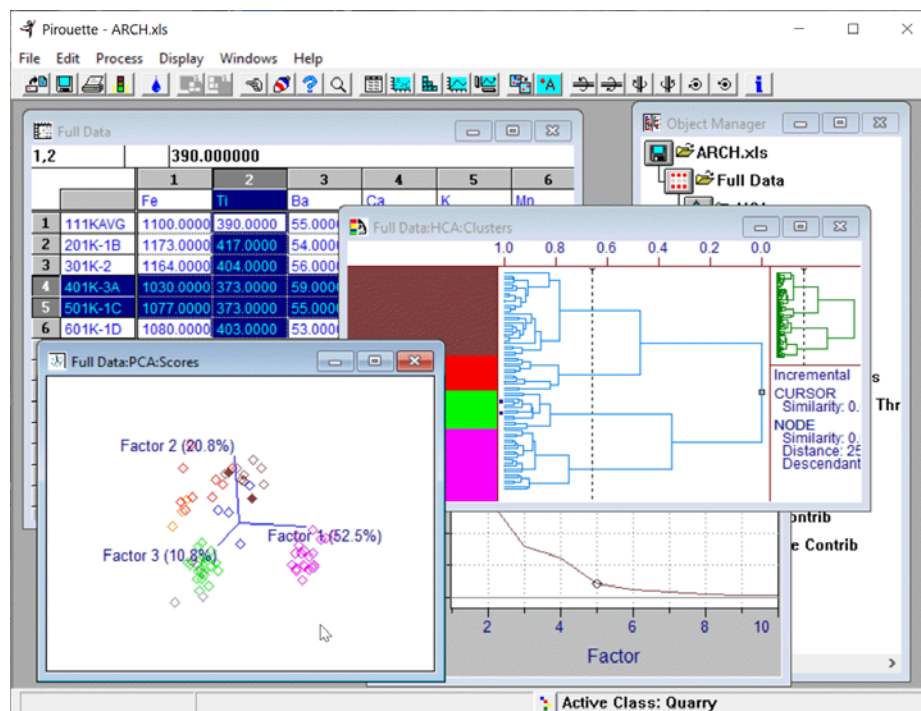
Where possible, the features and styles employed in Pirouette follow the standard adopted by modern Windows GUI programs. The Pirouette environment is organized around two sub-window styles: data-oriented graphs/tables and the Object Manager. Selections are made by pressing the left mouse button while the right mouse button performs special operations, such as displaying extra information or unmagnifying previously enlarged line and scatter plots. In keeping with both Pirouette and Windows tradition, menu options can also be accessed via keyboard equivalents.

A group of icons along the top edge of the Pirouette window, known as the ribbon, contains buttons to provide mouse access to common program features. These buttons are

1 Quick Start: Pirouette Briefly

grouped by specific function. The groupings include file and data processing functions, window manipulations, interaction tools, edit tools, view type buttons, plot tools and navigation aids.

Figure 1.1
The Pirouette environment



DATA INPUT

There are three ways to prepare data for analysis in Pirouette: by hand entering information, by pasting information from another application or by accessing an existing file with the Open Data item in the File menu.

Pirouette recognizes a variety of general and instrument-specific file formats. Common file formats are listed below. Other formats supported in the current version are discussed in “Other File Formats” on page 14-10.

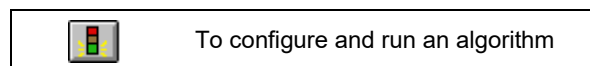
Table 1.1
Common Pirouette file types

Extension	Format Description
.PIR2	Pirouette’s native, fast loading binary format – which stores objects calculated during a Pirouette session
.DAT	An ASCII format which can be generated by a word processor or text editor – requires formatting specifiers (see “ASCII Files” on page 14-5)
.XLSX	The standard format created by Microsoft Excel – requires a few formatting specifics (see “Excel Files” on page 14-9)

RUNNING ALGORITHMS

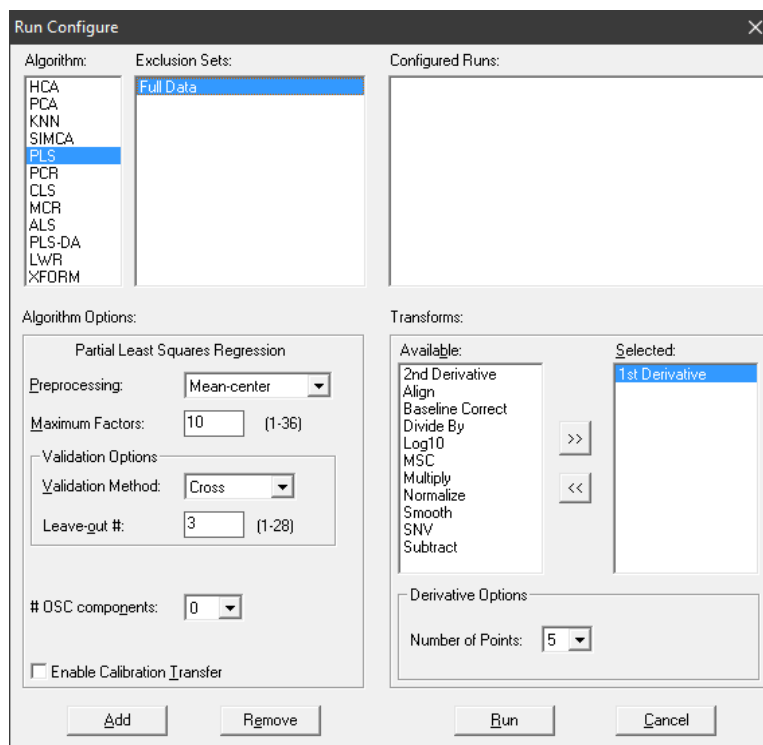
Once data have been input, multivariate analysis can begin. Clicking the Run button

Table 1.2
The Run setup button



brings you to the heart of Pirouette, a dialog box which presents a list of data subsets, transforms and algorithms with their associated options.

Figure 1.2
The Run Configure dialog box





Data analyses are configured by highlighting an entry in the Algorithm list box, selecting a data subset by highlighting an entry in the Exclusion Sets list box, then clicking on the Add button. Any modifications to Algorithm Options or Transforms must be made before the Algorithm/Exclusion Set combination is added to the configuration list. Note that Algorithm Options are algorithm-specific while Transforms can be applied to any algorithm. You can repeat the highlighting and adding process so that several algorithm/subset pairs are set into the list. Finally, click on the Run button to begin processing all configured algorithms.

VIEWING RESULTS








Once an algorithm run has finished, its results are made available via the Object Manager. Click and drag the algorithm folder to the Pirouette work area, and its results will be displayed in a single window containing an array of subplots, each showing one of the computed objects. The number of items in this window depends on the algorithm executed. You can interact with a subplot after zooming it to full window status. Any changes made to the plot in the zoomed state are maintained when it is unzoomed and returned to the array. Two buttons on the ribbon zoom and unzoom subplots.

Table 1.3
Ribbon buttons for manipulating array plots

Button	Description
	Zoom a subplot to the full window
	Unzoom a subplot (<i>i.e.</i> , back to its originating array)

Objects created by Pirouette take on one of the seven views shown in [Table 1.4](#). The first five views are generally available; the last two are algorithm specific. All views except the dendrogram are accessible from the ribbon; clicking a ribbon view button switches the view of the zoomed plot (that is, a plot not shown as an array).

Table 1.4
Pirouette's views

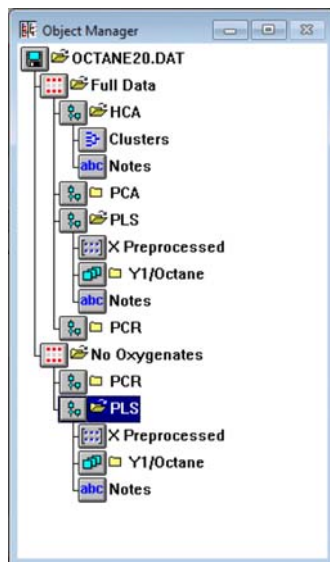
Icon	Description
	Table
	3D scatter plot
	2D scatter plot
	Line plot
	Multiplot
	Factor selection plot
	Dendrogram

In addition, a Notes window is produced for each algorithm. In this window are initially presented some information about the data processing steps used for that algorithm. This window is editable, and changes you make by typing are stored with the data, so it becomes a vehicle for recording commentary relevant to your analysis.

THE OBJECT MANAGER

Exclusion sets and computed results are organized via Pirouette's Object Manager, in essence a historian of file activity. The Object tree lists all existing raw and computed data objects. Tree items are iconic, revealing the structure represented (for details, see ["Object Manager icons"](#) on page 11-1). Note that subset and algorithm result names can be changed via the Rename item on the Objects menu.

Figure 1.3
Object Manager



New windows can be created via a procedure called “drag-and-drop”: click-drag an item from the Object Manager and drop it onto the work area. You can drag an algorithm result folder (to produce an array plot) or an individual item.

SAVING IMAGES

Pirouette provides three ways to store its images:

- Printing to a device or file
- Copying to the Clipboard
- Saving to a file

To capture graphics for a report, copy a bitmap or metafile image to the Windows clipboard and then paste it into another Windows application. To save an image to a metafile, select Edit/Copy Special/To File.

SAVING DATA AND MODELS

A data file can be saved in one of ten formats (Pirouette (legacy and current), ASCII (formatted, flat, and CSV), Excel (legacy and current), NWA (NQA), ChemStation (CH), Galactic (SPC) and ANDI (CDF)) and loaded back into Pirouette later or merged with another file of the same type. However, only the first format will preserve existing subsets and algorithm results. You may also save an Object Manager entity (*i.e.*, subset or computed object) into a file using File > Save Objects.

Note: Starting with Pirouette 4.0 rev 1, the ability to run as a standard User has been enabled. However, such a User with limited permissions cannot write to folders in Program Files, the default path for Pirouette. Instead, save your files in My Documents or a sub-folder thereof.

Prediction models from all algorithms except HCA can be saved for later use by Pirouette or InStep. When a model is saved using File > Save Models, the Pirouette binary format is the default. SIMCA, KNN, PLS and PCR models can also be saved in ASCII and other formats. See “ASCII Models” in Chapter 15 for details.

PIROUETTE HELP

This version of Pirouette includes extensive on-line documentation to assist you in learning about Pirouette and multivariate data analysis. Help has been implemented by converting the Pirouette manual to Adobe's portable document format (PDF), which is suitable for presentation of information on almost any computer platform. Pirouette's PDF files can be viewed with Acrobat Reader, which is readily available and integrated into most browsers. You may use Acrobat itself or a web browser of your choice so long as the Acrobat Reader plug-in has been included with the browser as described in ["Setup" on page 16-45](#).

Help contains context-sensitive hyper-links, an index and a main user guide document with built-in chapter and section bookmarks to facilitate navigation. Acrobat or your browser can be opened directly from the Pirouette Help menu. When referring frequently to Help, leave the browser open in the background and switch to it using Alt-Tab. Because the PDF format works on most platforms, Help documents can be moved to another platform (*e.g.*, Unix, Macintosh) without any modification.

Technical Support

Pirouette is a premium product, and with it Infometrix offers readily available technical support. We can assist if you are having difficulty in installing or running the software. If you have questions about the use of the technology to solve specific problems, Infometrix also provides consulting and/or training. Applications information and links to other chemometric sites are available on our web page. Feel free to contact us (see ["Technical Assistance" on page 18-14](#)) for more details.

Pattern Recognition Tutorial

Contents

The Basics	2-1
Exploratory Analysis	2-7
Modeling and Model Validation	2-18
Review	2-23

This chapter introduces Pirouette by working through a multivariate pattern recognition example. Multivariate analyses should begin by defining the problem and assuring the validity of the data. Thus, an exploratory analysis should always be performed even if you intend to develop models from the data set. If the results of this exploration indicate that the data are appropriate for building a classification model, then one of the Pirouette's classification algorithms can be used to group samples into categories.

This tutorial is based on a well-described data set which contains elemental composition of both obsidian quarry samples and obsidian artifacts which may have originated from the quarries¹. The goal is to teach you not only the Pirouette interface but also chemometric fundamentals. To present the material as a 30 to 60 minute session, the chemometric interpretation is necessarily light. For additional detail on interpreting algorithmic results, refer to [Part II Guide to Multivariate Analysis](#).

The Basics

DEFINE THE PROBLEM

The first step is to establish the purpose of the investigation: for this data set, it is to determine if the quarry of origin can be identified for the artifacts. If so, it might be possible to assess migration patterns and trading routes of the indigenous cultures using these tools. Although drawn from archaeology, the example is in fact a generic pattern problem: to classify samples into categories. These categories might be based on geographic or manufacturing origin or might relate to product category (*i.e.*, good or bad). Typical general questions include:

- Is the analytical data appropriate for classifying samples?
- Can we determine the category of a sample from its chemical composition?
- How reliable are the classifications we develop?

2 Pattern Recognition Tutorial: The Basics

The data are X-ray fluorescence determinations of ten trace metals in quarry samples and artifacts. The four quarries are north of the San Francisco Bay area and the artifacts were found in several neighboring locations¹. Specific questions include:

- Is the trace metal signature of each quarry sufficiently different to distinguish among them?
- How homogeneous are the samples drawn from each quarry?
- Do artifacts have trace metal signatures similar to those of the quarries?

OPEN THE FILE

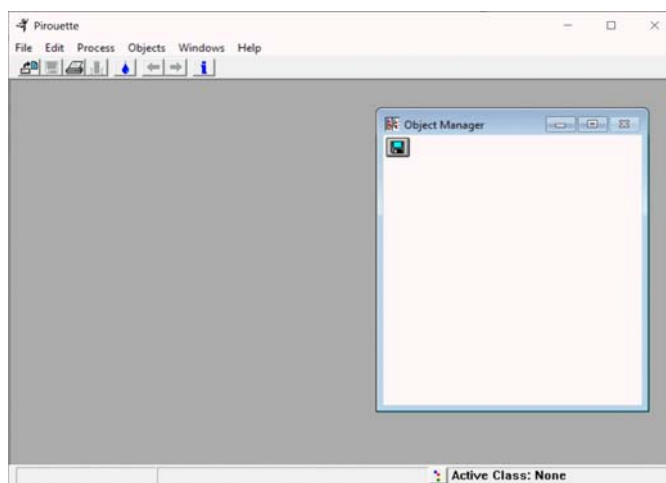
To begin using the program,

- click on START, select Programs, the Infometrix folder, the Pirouette folder, then the program icon for Pirouette

The screen will appear as shown in the figure below. Note the menu commands and button ribbon along the top and the Object Manager window. Go to Windows > Preferences > Chart > Window Attributes and change the Maximum Number of Windows Created value to 2. Click on OK to close the dialog box.

Note: You can change this value back to 0, the default, after the exercise.

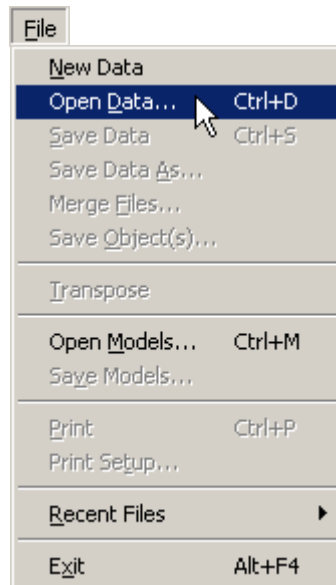
Figure 2.1
Pirouette's start up screen




The tutorial data are stored in a file called ARCH.XLS which is supplied with Pirouette. To load a data file:

- Click on the File menu at the top of the Pirouette window
- Move down to the Open Data... item as shown below

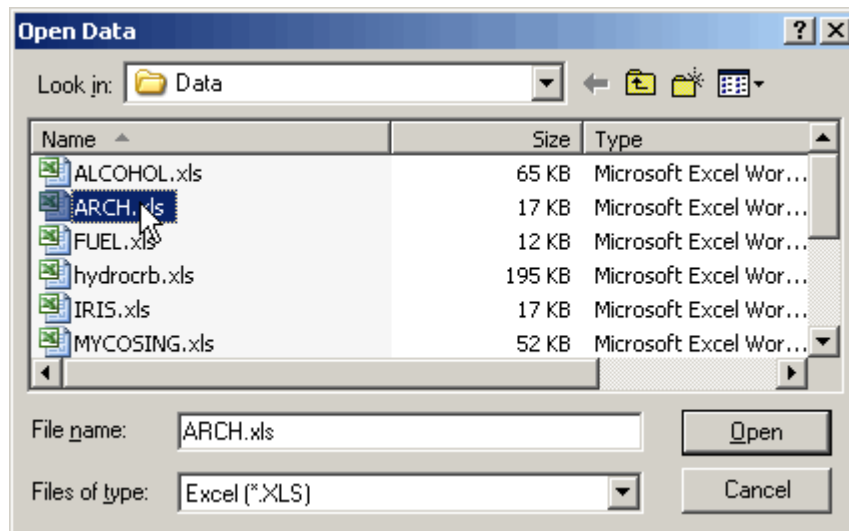
Figure 2.2
Choosing Open Data
from the File menu



Note: The leftmost button on the Ribbon also accesses the Open Data dialog box. 

The Open Data dialog box shown below allows you to select drives, navigate their directory structure, and filter by type the list of files displayed.

Figure 2.3
The Open Data
dialog



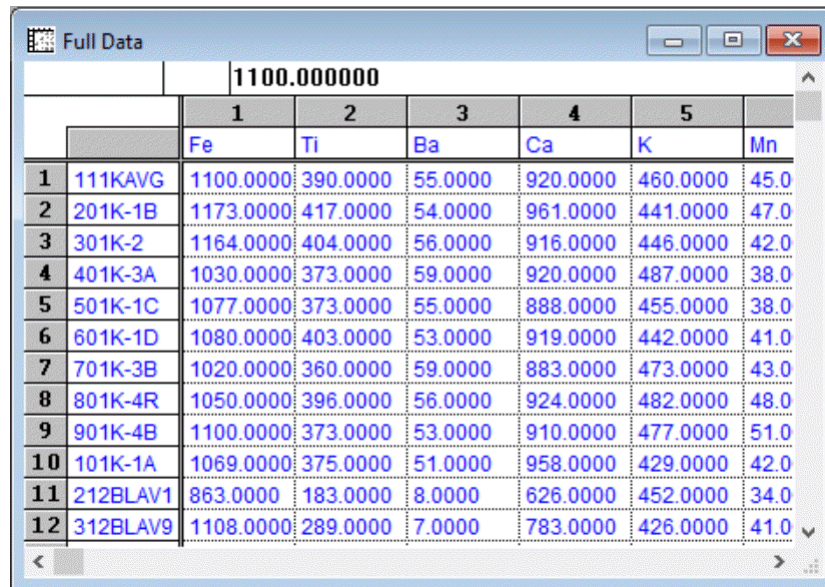
In this example, ARCH . XLS resides in the Data directory which is itself in the Pirouette directory. Once you have moved to the DATA directory and specified Excel in the Files of Type box,

- Highlight the file name by clicking on it (as in the above figure)
- Click on Open

and ARCH . XLS is loaded. Click on the Full Data entry in the Object Manager and, while pressing the left mouse button, drag the mouse cursor outside of the Object Manager win-

down until the cursor changes form. Release the mouse button and the ARCH data are presented as a table view.

Figure 2.4
The ARCH data in a
table view




		1	2	3	4	5	
		Fe	Ti	Ba	Ca	K	Mn
1	111KAVG	1100.0000	390.0000	55.0000	920.0000	460.0000	45.0
2	201K-1B	1173.0000	417.0000	54.0000	961.0000	441.0000	47.0
3	301K-2	1164.0000	404.0000	56.0000	916.0000	446.0000	42.0
4	401K-3A	1030.0000	373.0000	59.0000	920.0000	487.0000	38.0
5	501K-1C	1077.0000	373.0000	55.0000	888.0000	455.0000	38.0
6	601K-1D	1080.0000	403.0000	53.0000	919.0000	442.0000	41.0
7	701K-3B	1020.0000	360.0000	59.0000	883.0000	473.0000	43.0
8	801K-4R	1050.0000	396.0000	56.0000	924.0000	482.0000	48.0
9	901K-4B	1100.0000	373.0000	53.0000	910.0000	477.0000	51.0
10	101K-1A	1069.0000	375.0000	51.0000	958.0000	429.0000	42.0
11	212BLAV1	863.0000	183.0000	8.0000	626.0000	452.0000	34.0
12	312BLAV9	1108.0000	289.0000	7.0000	783.0000	426.0000	41.0

EXAMINE THE DATA

Scan the table Full Data to get a feel for its general structure: ten columns of trace metal measurements, which are independent variables, and an eleventh column (C1) named Quarry, which is a categorical (or class) variable. Quarry values 1, 2, 3 and 4 identify samples from the four obsidian source sites, while values 5, 6 and 7 are assigned to artifact samples. Scanning vertically, you can determine that the table contains 75 cases (rows) of which 63 are quarry samples and 12 are artifacts. Quarry samples names begin with three digits while artifact names begin with the letter *s*.

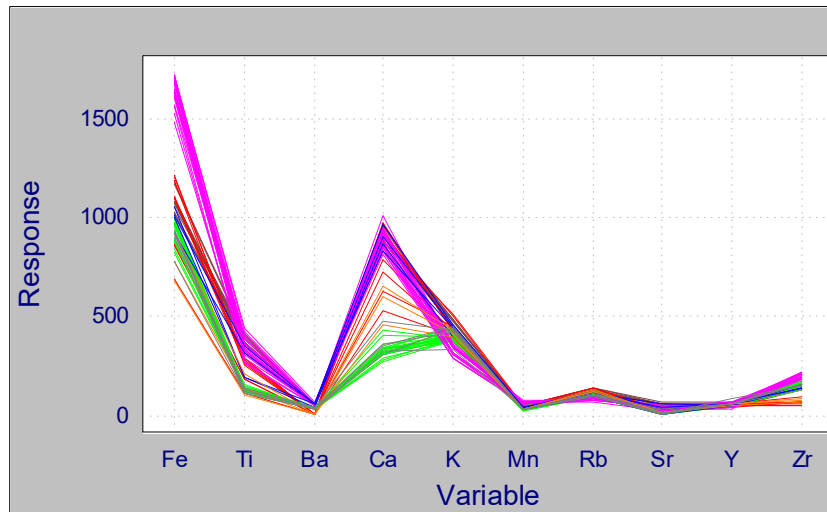
Note: The name of the class variable which is active appears in the status bar message area at the bottom of the Pirouette window. For ARCH, the message reads "Active Class: Quarry".

Line plotting the data is always advisable to locate obvious outliers and to decide if transforms and /or preprocessing will be appropriate.

- Click on the Line Plot button 

A trace's color is mapped to the class variable value. It is immediately apparent from the following figure that there are large values and large relative variances for iron (Fe, Var. #1) and calcium (Ca, Var. #4). This observation will be important later when we choose a preprocessing method. No obvious outliers are apparent, so we can start exploratory analysis without first excluding any samples.

Figure 2.5
The ARCH data as a
line plot



It is advisable to examine the data using other graphic tools. To see a series of variable biplots as shown in the next figure:


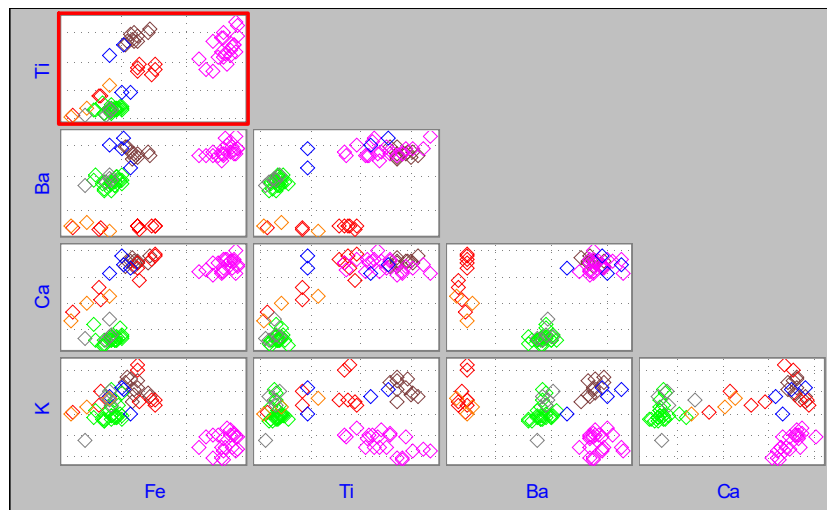
- Click on the Multiplot button 

Figure 2.6
The multiplot view

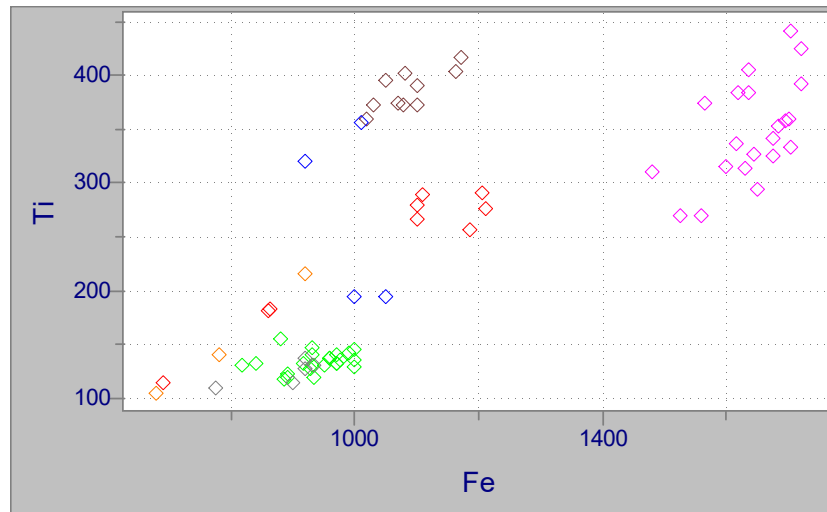


To see the expanded version of titanium plotted against iron (Ti vs. Fe) shown in the next figure,

- Double-click on the outlined plot in the upper left corner

2 Pattern Recognition Tutorial: The Basics

Figure 2.7
2D View of Ti vs. Fe



To see a 3D plot of Ti, Fe and Ba,


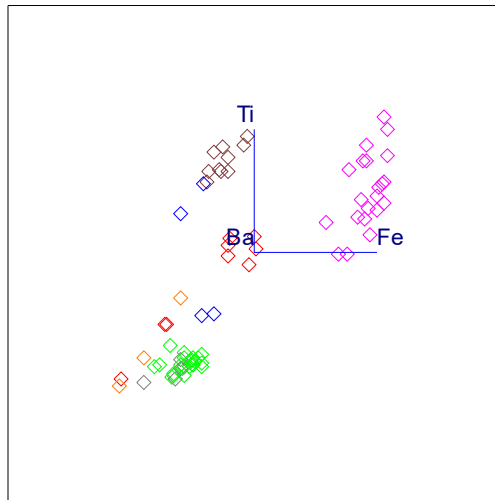

- Click on the 3D button 

Figure 2.8
A rotatable 3D plot of
Full Data



The default cursor for the 3D view looks like a top. To rotate the plot,

- Position the top cursor  over the plot area
- Move the mouse while pressing the left button

Use all of these views to investigate groupings in the data and look for trends. In the case of ARCH, clustering by quarry is evident in the raw data.

- Click on the Table button  to return to a tabular view

Exploratory Analysis

Now we'll perform an exploratory analysis using two complementary methods, Hierarchical Cluster Analysis (HCA) and Principal Component Analysis (PCA).

RUNNING EXPLORATORY ALGORITHMS

To initiate an exploratory analysis,

- Click on the Process menu and select the Run item

The Run Configure dialog box will open, showing available algorithms and an exclusion set. In this case, only the Full Data subset exists, which is automatically highlighted. For a complete description of exclusion sets, see [“Subsets” on page 11-9](#).

Note: The Run button also opens the Run Configure dialog box.



To visualize the relationships among samples, we will select HCA and PCA. The difference in magnitude of the responses for the elements noted earlier suggests Autoscale preprocessing. For an explanation of this choice, see [“Preprocessing” on page 4-26](#).

To configure an HCA run:

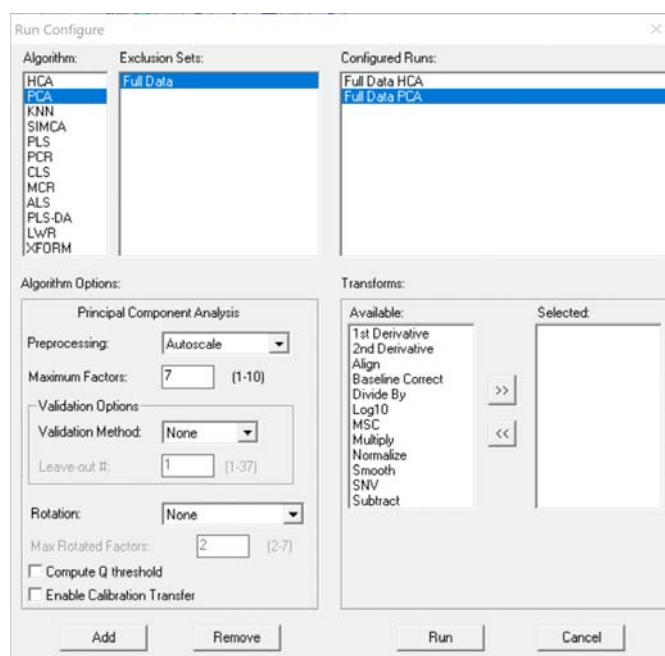
- Click on HCA in the algorithm list
- Click on the down arrow in the box to the right of Preprocessing and select Autoscale
- In a similar fashion, choose Incremental Link as the Linkage Method
- Click on Add at the bottom left of the dialog box

To configure a PCA run:

- Click on PCA in the algorithm list
- Change Preprocessing to Autoscale as in HCA above
- Click on Maximum Factors and change the number to 7
- Click on Add at the bottom left of the dialog box

The two items in the Run Configuration box show both the exclusion set and the algorithm to be applied to it: Full Data HCA and Full Data PCA. When you have finished setting up the run, the dialog box should appear as follows.

Figure 2.9
The Run Configure
dialog



To start processing,

- Click on Run at the bottom of the dialog box

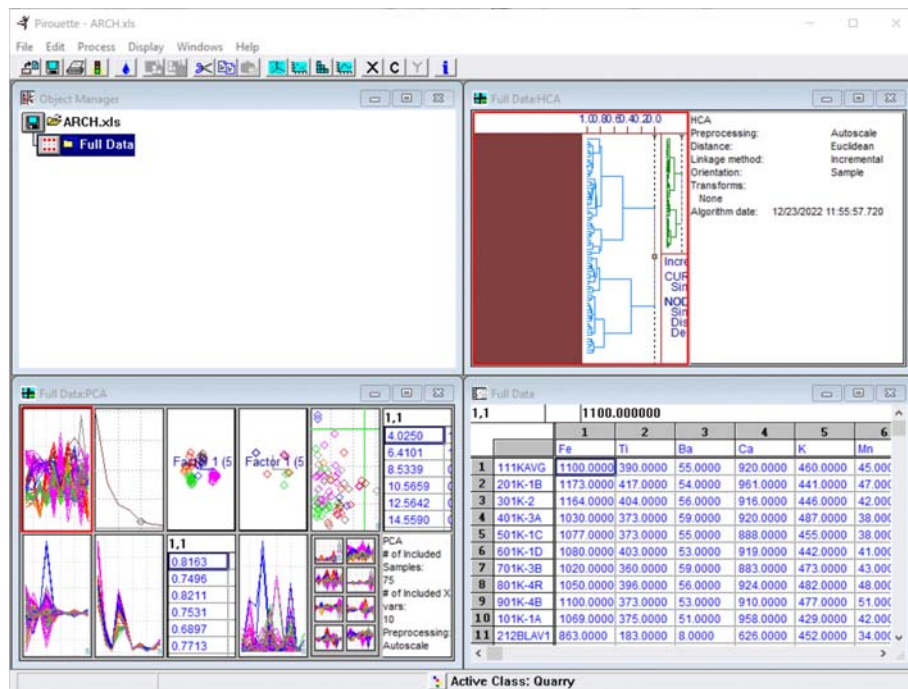
While calculations are performed, a Run Status dialog box is displayed. As each run finishes, a results window is presented if your Windows > Preferences > Chart > Window Attributes > Maximum Number of Windows Created is set to 2 (0 is the default). If no problems are encountered during processing, the Run Status box closes when all runs have completed. Otherwise, it remains open; the condition which caused a run to abort is described when the Details button is clicked.

Four windows are now available with the Object Manager showing an iconic representation of every computed result. The remaining three windows contain Full Data, Full Data HCA results and Full Data PCA results. To see what we have at this point,

- Select the Tile item from the Windows menu.

A plot similar to the one shown below should appear.

Figure 2.10
Tiled HCA and PCA
results

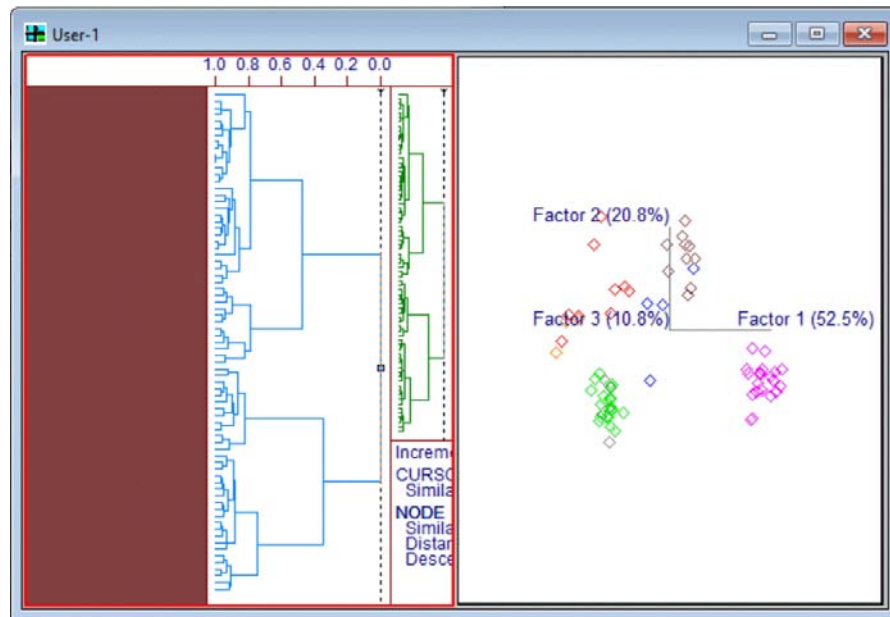


We now focus on two results and utilize Pirouette’s drag and drop capabilities to create a custom chart. First, close the HCA and PCA windows by clicking on the go-away box. This leaves the Full Data and Object Manager windows open. To make a custom plot,

- Click on the Object Manager window to make it active
- Double-click on the Full Data folder to show which algorithms have been run
- Double-click on the HCA folder to reveal its computed objects
- Double-click on the PCA folder to reveal its computed objects
- Click on the Clusters icon to select it
- With the Ctrl key held down, click on the Scores icon to select it as well
- Keeping the mouse button depressed, drag over empty space in the Pirouette window
- Release the mouse button to complete the drag and drop

The screen appears as below.


Figure 2.11
Customized HCA
dendrogram and
PCA Scores view



During the drag and drop, the cursor changes to a dragging tool as it begins to move. As the cursor moves over the open space, it again changes form to indicate that dropping is allowed. On release of the mouse button, the window redraws with the two graphics displayed side-by-side; the window is titled User to reflect its custom status. You can use this method to create any number of custom plot arrays.

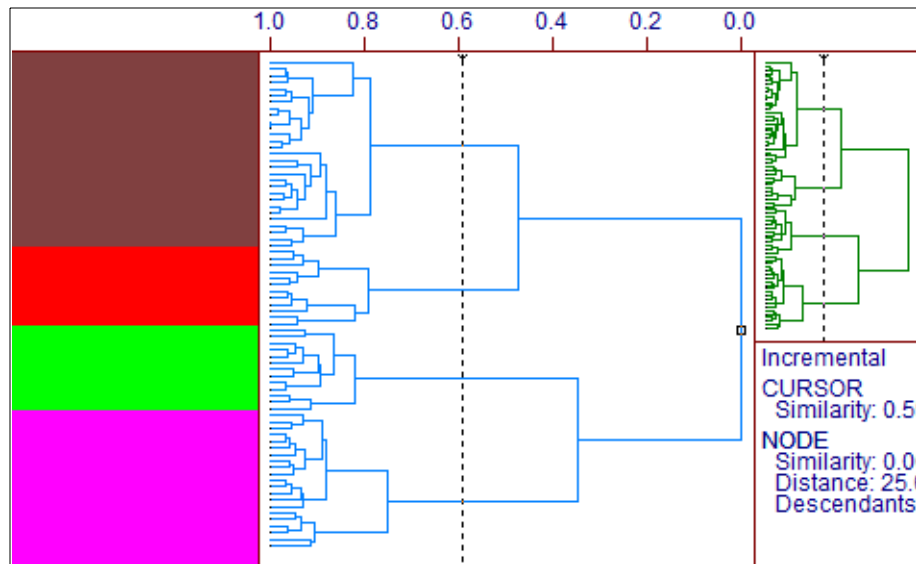
DATA INTERPRETATION


To check on the distinct clusters found in our initial examination of Full Data,

- Click on the dendrogram to make it the active subplot
- Expand it to full window with the Zoom button 
- Position the cursor over the vertical dashed line at its top end where it becomes a double arrow
- Click-drag the line to the left to a cursor similarity of about 0.60
- Release the mouse button

This defines and colors four clusters in the dendrogram. The result is shown below.

Figure 2.12
ARCH dendrogram
with a cursor
similarity of 0.60



- Click the Unzoom button to shrink the dendrogram and produce a window which again contains two subplots 


To expand the 3D scores plot to full window,

- Click on the Scores subplot
- Click on the Zoom button

Note: The Zoom and the Unzoom buttons are used to navigate through array plots and multiplots. The Zoom button acts on the plot which is surrounded by a thicker border (red by default). It has mouse and keyboard equivalents: double-click on the subplot or press Enter. Similarly, unzoom a plot by shift-double-clicking or pressing Ctrl-Enter.




To display the scores as a 2D plot with points labeled by sample index,

- Click on the 2D button 
- On the Display menu, choose the Point Labels item, then the Index item
- Click on the Unzoom button to shrink the 2D plot

Your view should look similar to Figure 2.15. Note the ellipse surrounding the points on the Scores display; this is the 95% confidence interval for the full ARCH dataset.

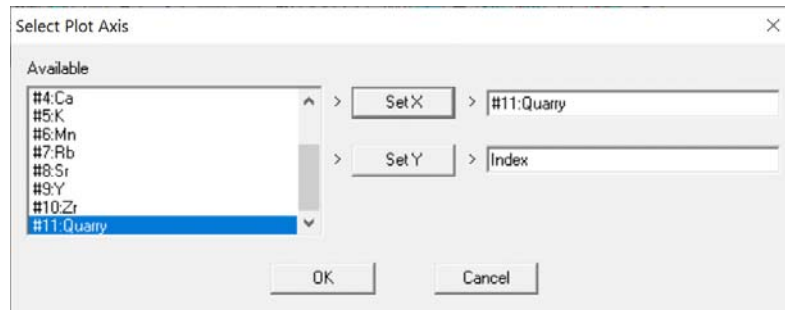
Next, we will setup a view of Full Data which allows strategic selection/highlighting of particular samples. To organize samples by Quarry value,

- Click on the Full Data window to make it active
- Click on the 2D button to convert the table view to a scatter plot
- Click on the Selector button  to see its dialog box (shown below)
- Select Sample Index and click on the Set Y button
- Scroll down the Available list until Quarry is displayed


2 Pattern Recognition Tutorial: Exploratory Analysis

- Select Quarry and click on the Set X button
- Click on OK

Figure 2.13
The Selector dialog

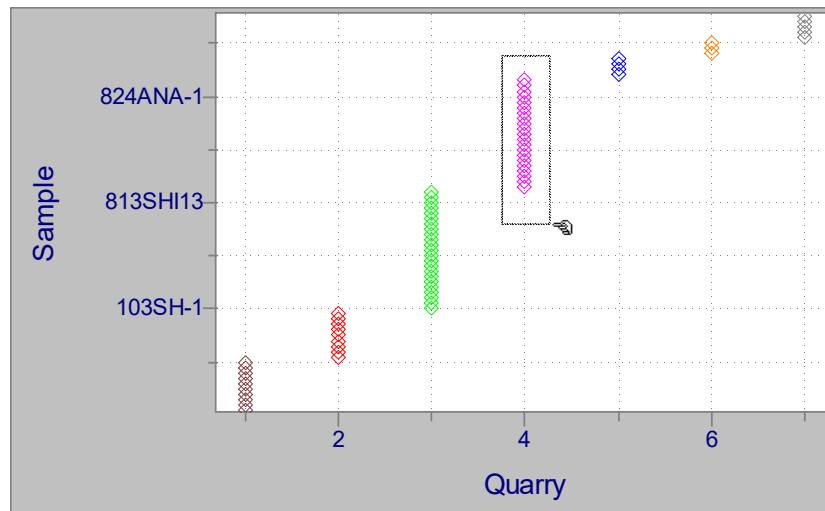


The 2D plot now shows Quarry category versus sample number. Such a view allows us to easily select/highlight all members of specific groups. To select all Quarry 4 samples,

- Click and drag a box around them, using the Pointer 

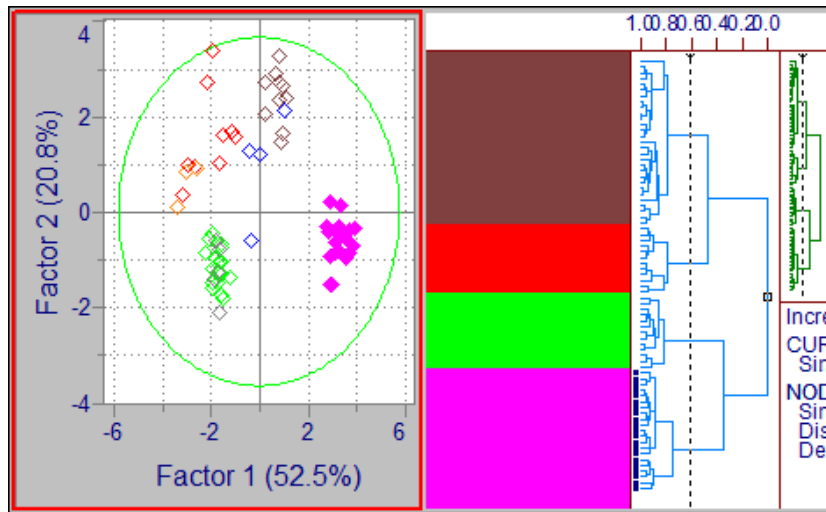
The result is shown in the following figure.

Figure 2.14
Highlighting Quarry
4 samples



Highlighting is manifested by filled points so that those selected can be differentiated from unselected, unfilled ones. This highlighting is mapped to every other relevant display, whether graphical or tabular. Therefore, the Quarry 4 samples appear highlighted in the previously created User chart containing the dendrogram and scores. These samples occupy the bottom branch in the dendrogram and cluster in a localized region on the right of the scores plot.

Figure 2.15
Highlighted Quarry 4
samples in the HCA
and PCA views



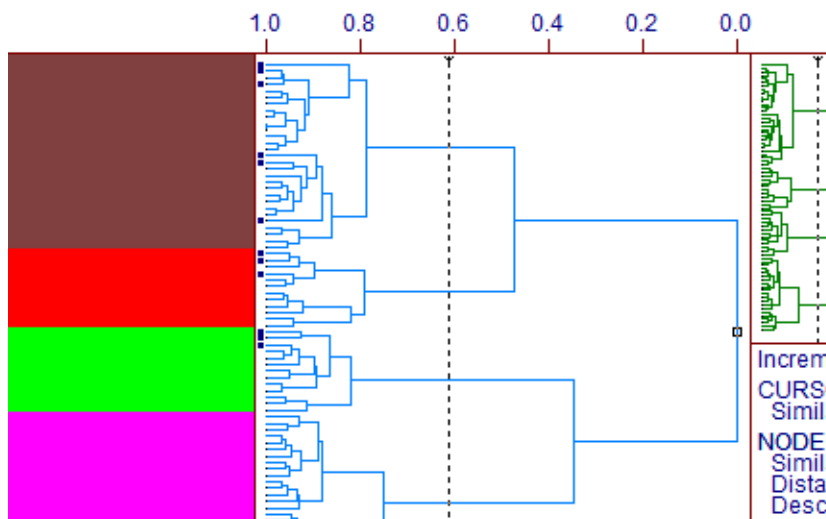
We can locate Quarry 1, Quarry 2 and Quarry 3 members in a similar fashion by highlighting them in the Quarry vs. Sample # plot and then examining the dendrogram and scores plot. We find that each quarry occupies a separate region of both the dendrogram and scores plot which implies that the trace metal signature of each quarry can be distinguished. It is thus probable that a successful classification model can be built from the ARCH data.

We can also highlight the artifacts (the so-called “Quarries” 5, 6 and 7) and see that they are located on dendrogram branches associated with Quarries 1, 2 and 3, but none on the Quarry 4 branch.

- Zoom the dendrogram to full window

so that the view looks like that in the following figure.

Figure 2.16
No artifacts in the
Quarry 4 cluster

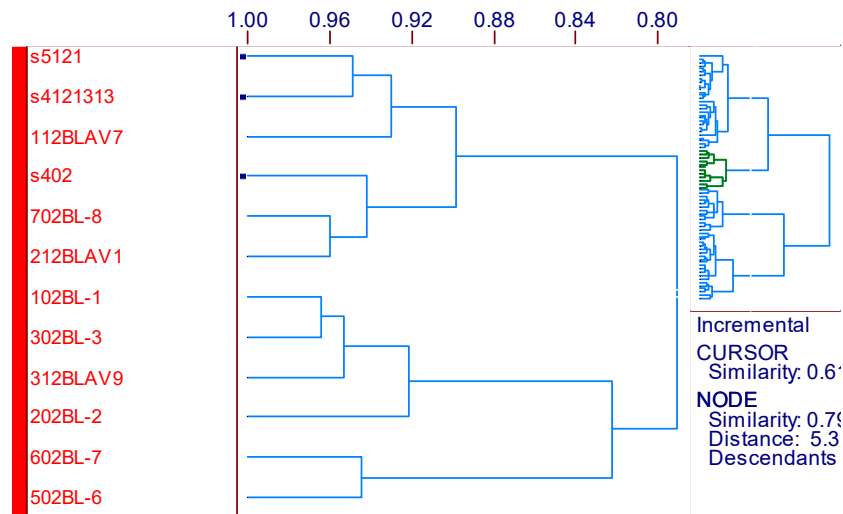


In effect, this dendrogram classifies the artifacts. To see which artifacts cluster with Quarry 2,

- Move the cursor to the position shown in the figure above
- Double-click that node to produce the next figure

2 Pattern Recognition Tutorial: Exploratory Analysis

Figure 2.17
The dendrogram
expanded



All branches to the left of the node just specified now fill the left side of the window. The miniature dendrogram on the upper right, called the overview, shows the expanded region in a different color. Double-clicking on nodes in either the expanded or overview regions is an easy way to navigate the dendrogram. You can step out of the expanded view one node at a time by clicking on the far right of the expanded dendrogram when the cursor takes on a right arrow shape.

Clicking on a dendrogram node marks it with a small circle in the expanded region. Node information is presented in the lower right portion of the dendrogram window.

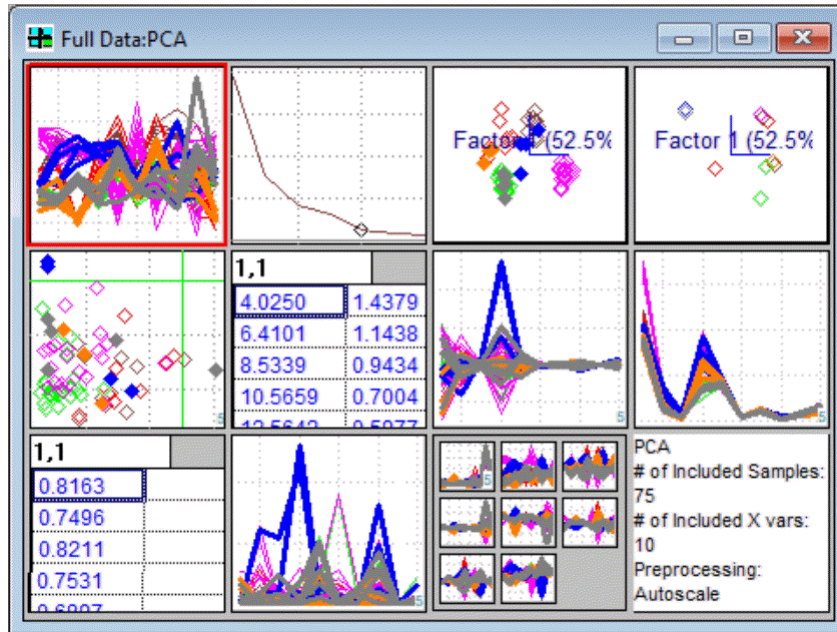
Note: *The dendrogram view cannot be converted to any other view. For that reason, no view switching buttons are available when the HCA window is active. Also, whenever a small enough number of samples appear in the expanded region, names replace the color bar on the far left. Making the window taller allows the display of more names.*

When sample names show in the dendrogram, you can see how the artifacts match against specific quarries.

To understand a little more about the ARCH data prior to performing classification modeling,

- Click on the PCA folder
- With the left mouse button down, drag the PCA folder to a blank portion of the work area
- Drop the folder by releasing the mouse button

Figure 2.18
PCA results



PCA results are presented as an array of subplots. Subplots 2-4 are basic PCA objects, while the next six are modeling diagnostics. In this walkthrough, we are concerned with exploratory analysis and so address only the basic PCA objects. Other objects have SIM-CA analogs which are discussed in “Soft Independent Modeling of Class Analogy” on page 6-15.

The second subplot describes how much of the total variance in the ARCH data is explained by each additional principal component. To see the variance values,



- Double-click on the second subplot to zoom it to full window
- Click on the Table button. 

Figure 2.19
Table view of the
PCA Factor Select
object

		1	2	3	4
		Variance	Percent	Cumulative	Press Cal
1	Factor1	388.645	52.520	52.520	351.355
2	Factor2	153.804	20.784	73.304	197.551
3	Factor3	80.033	10.815	84.119	117.519
4	Factor4	62.026	8.382	92.501	55.493
5	Factor5	21.826	2.949	95.450	33.667
6	Factor6	12.908	1.744	97.195	20.759
7	Factor7	8.913	1.204	98.399	11.846

The first principal component explains over 52% of the total variance; more than 95% of the variance is captured by the first 5 principal components.

- Click the Unzoom button to return to the array plot 

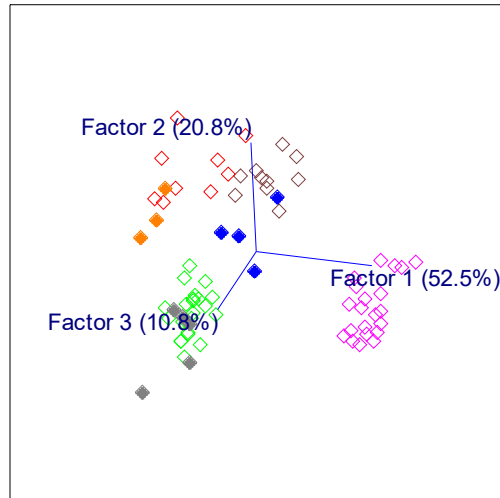
The third and fourth subplots (shown as 3D views) are the scores and loadings. You can think of the principal component axes as representing a compressed view of the multi-


2 Pattern Recognition Tutorial: Exploratory Analysis

variate data. The scores are a mapping of the original sample data onto a coordinate system specified by the loadings. To see the relationship among the samples,

- Double-click on the third subplot to expand it to full window
- Click-drag the mouse from the NE to the SW to get a view similar to that shown in the following figure, where point labels are displayed as their row numbers

Figure 2.20
Rotated ARCH
scores



- Click on the ID button 
- Position the question mark cursor over the point in the extreme SW
- Click and hold to show the sample number and name, #75: s5136953

Sample #75, an artifact because its name starts with the letter *s*, is similar to the green Quarry 3 samples except for its larger coordinate on Factor3. Sample #75 appears somewhat separate in the scores plot and may require further investigation. To produce a figure like the one below,


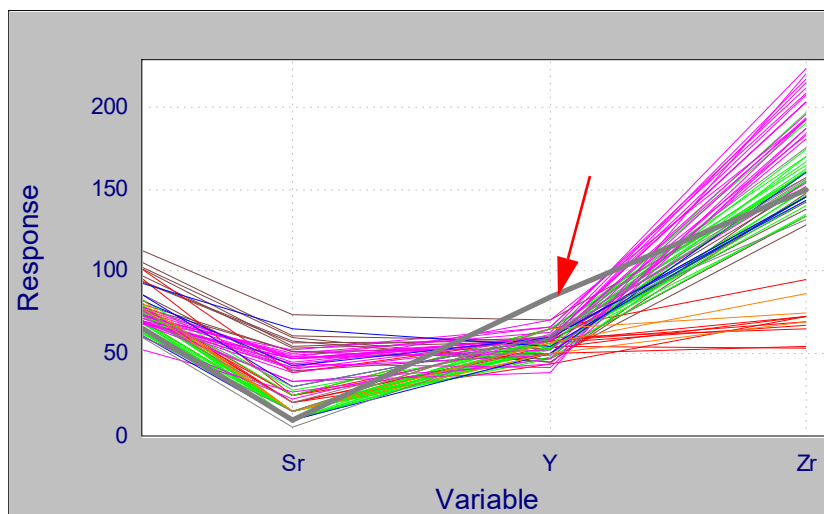
- Select this point with the Pointer tool
- Drag and drop Full Data from the Object Manager and switch to a line plot view
- Click on the Magnify button 
- Click-drag in the plot region around Sr and Zr

Figure 2.21
A magnified line plot
of Full Data

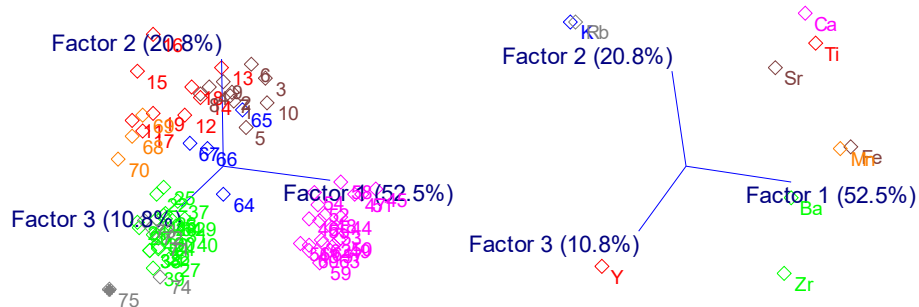


Thus, sample #75 is distinguished from the Quarry 3 (and all other) samples by an anomalously large Yttrium concentration shown by the highlighted gray trace in the above figure.

- In the Object Manager, open the PCA folder
- Drag and drop the Loadings object onto an open portion of the Pirouette desktop
- Go to the Display menu and select Point Labels\Name to label show the element assignments.

The result should look like the plot shown below. By rotating the loadings axes to a position similar to the Scores view, we can see that displacement of sample 75 along the Factor 3 axis is indeed related to Yttrium (Y) content, also displaced along the Factor 3 axis in the loadings plot. Looking at scores and loadings plots side-by-side will illustrate this correlation.

Figure 2.22
Loading plot
identifies the reason
for the #75 outlier



Before designating sample #75 an outlier, we should confirm its yttrium measurement and decide if the variability in Quarry 3 is well-represented. Then we could use classification methods such as KNN and SIMCA to see if they corroborate the assignments implied by the HCA and PCA clustering.

Modeling and Model Validation

Exploratory analysis with HCA and PCA has shown that there are separate clusters of samples, indicating that analysis with pattern recognition algorithms ought to succeed. Thus, the next stage of analysis will be to build models that can be used to predict from which category (i.e., quarry) a sample originates.

KNN and SIMCA are algorithms which build models that classify samples into discrete categories. Both are based on the concept of proximity, the assumption that if a set of measurements for an unknown sample is very similar to that of a specific group, then the unknown is likely to be a member of that group. KNN classifies based on a plurality vote of a specified number of nearest neighbor known samples. SIMCA finds principal component axes in the multivariate measurement space for each category. An unknown is classified as belonging to a specific group if it lies “closest” to the group and within an allowed threshold. For a comprehensive discussion of KNN and SIMCA, refer to [Chapter 6, Classification Methods](#).

KNN MODELING

K-Nearest Neighbor performs its decision making by computing the distance of each sample to all the samples in the data matrix. If the distances are ranked in ascending order, we can examine the list to determine which samples are closest to the sample being analyzed. Each sample in the matrix also carries a category identification. If we decide to look only at the nearest samples (i.e., the one with the smallest distance), then we can say that our test samples is most likely a member of the category of the sample that is closest. But, if we compare the test sample to the neighbors with the k smallest distances, the situation is a little more complicated: each sample contributes its vote for its category. Thus, for 3NN (the 3 samples of smallest distance), we will consider only the categories for the 3 nearest samples. The category with the most votes is that which is assigned to the test sample.

Let's try this with our archaeological data set. First, however, we will make two subsets. The first is composed of the set of samples taken from known quarry sites while the second is a set of artifact samples whose rocks assumedly were collected from one of these quarries. Therefore, we will want to make our classification model from only the quarry samples to later attempt to classify the artifact samples according to their origin.

- Select Windows > Close All Windows
- Drag Full Data onto the work area.

Scroll down so that samples below row 60 are showing.

- Click to select row 64
- With the Shift key held down, click again on row 75

All of the rows from 64 to 75 should be selected as shown in the figure below.

Figure 2.23
Selecting artifact
samples to exclude

Full Data		1050.000000					
		1	2	3	4	5	6
		Fe	Ti	Ba	Ca	K	Mn
63	124ANA-2	1600.0000	316.0000	54.0000	915.0000	347.0000	83.0000
64	s2112909	1050.0000	195.0000	46.0000	865.0000	400.0000	36.0000
65	s3111309	1010.0000	357.0000	65.0000	900.0000	455.0000	42.0000
66	s4111313	920.0000	320.0000	60.0000	830.0000	440.0000	33.0000
67	s5116953	1000.0000	194.0000	58.0000	965.0000	460.0000	42.0000
68	s402	920.0000	215.0000	6.0000	650.0000	435.0000	37.0000
69	s4121313	780.0000	140.0000	12.0000	605.0000	415.0000	35.0000
70	s5121	680.0000	105.0000	10.0000	460.0000	400.0000	31.0000
71	s1132909	920.0000	127.0000	39.0000	475.0000	430.0000	34.0000
72	s2132910	900.0000	115.0000	37.0000	310.0000	440.0000	34.0000
73	s3132910	920.0000	138.0000	41.0000	350.0000	450.0000	34.0000
74	s4136953	775.0000	110.0000	35.0000	327.0000	337.0000	34.0000
75	s5136953	935.0000	131.0000	38.0000	360.0000	420.0000	37.0000
76							

The selected samples are the artifacts. All of the samples above row 64 are from the quarries.

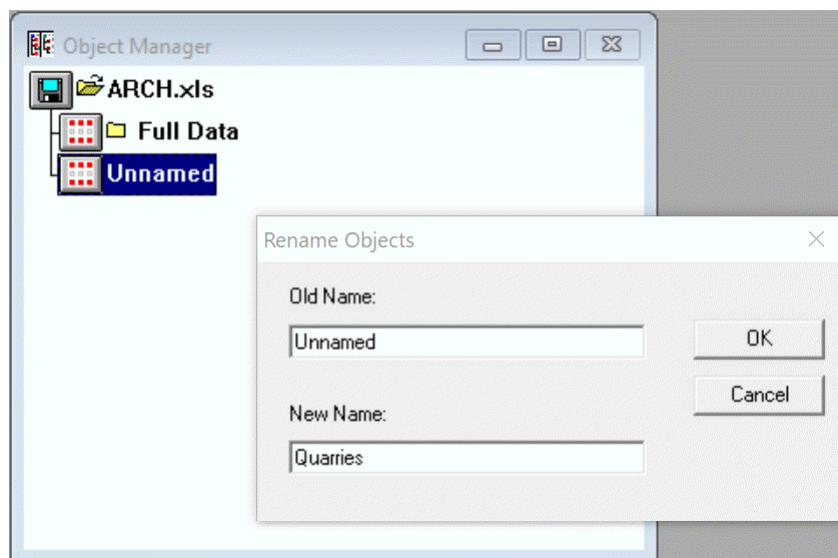
- Select Edit > Create Exclude

This creates a new subset with the highlighted rows excluded, retaining only the quarry samples as included. Pirouette supplies a default name for the subset of Unnamed. It's a good idea to name your subsets after their creation so you can remember what your purpose was.

- Click on the Unnamed subset in the Object Manager window.
- Select the menu item Objects > Rename

In the dialog box that is shown, enter a new name for the subset, for example, Quarries, as shown below.

Figure 2.24
Renaming new
subset



2 Pattern Recognition Tutorial: Modeling and Model Validation

While we're at it, let's make a subset containing only the artifact samples.

- Drag Full Data onto the work area again
- Click on Sample 1 index to highlight the first row.
- Scroll down and shift-click on row 63.

This should highlight all rows from 1 to 63, the quarry samples.

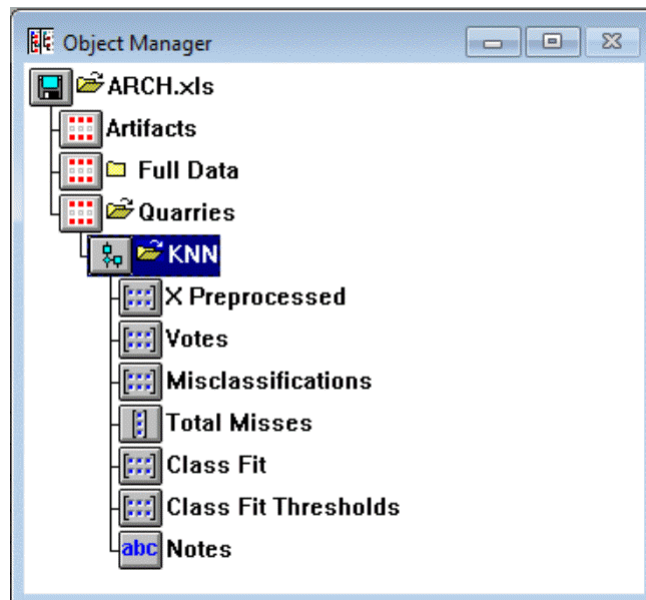
- Select Edit > Create Exclude
- Click on Unnamed in the Object Manager
- Select Objects > Rename again

Now, type in a name for this subset, such as Artifacts. Now we're ready to create our classification model.

- Select Process > Run
- Choose KNN from the list of algorithms
- Choose Quarry from the list of Exclusion Sets

We will want to use the same preprocessing parameters as before (autoscale), then hit Run to start processing. A Run Status dialog box will appear briefly. Then, in the Object Manager, the Quarry subset will show a folder icon to indicate that results are present. Opening the folders reveals the objects computed during KNN, as in the following figure.

Figure 2.25
Showing KNN result
objects



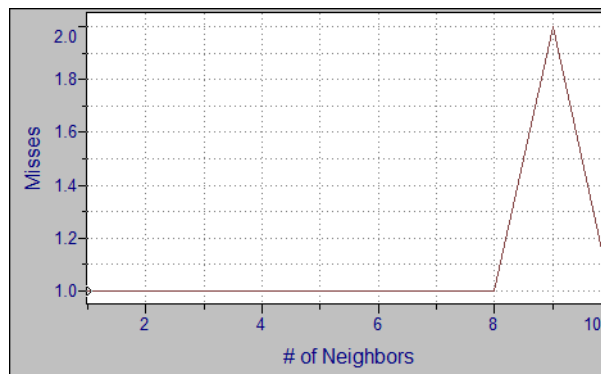
Drag the Votes object into the work area. This table contains all of the results of the KNN analysis: for each sample, the predicted category at every value K neighbors, in this case up to 10 neighbors.

Figure 2.26
Votes matrix

		1	2	3	4	5	6
		K=1	K=2	K=3	K=4	K=5	K=6
1,1		1.000000					
1	111KAVG	1	1	1	1	1	1
2	201K-1B	1	1	1	1	1	1
3	301K-2	1	1	1	1	1	1
4	401K-3A	1	1	1	1	1	1
5	501K-1C	1	1	1	1	1	1
6	601K-1D	1	1	1	1	1	1
7	701K-3B	1	1	1	1	1	1
8	801K-4R	1	1	1	1	1	1
9	901K-4B	1	1	1	1	1	1
10	101K-1A	1	1	1	1	1	1
11	212BLAV1	2	2	2	2	2	2
12	312BLAV9	2	2	2	2	2	2
13	202BL-2	2	2	2	2	2	2
14	302BL-3	2	2	2	2	2	2

To establish a KNN model, we need to decide how many neighbors are appropriate. This would be tedious to do with the Votes table; instead drag the Total Misses object to the work area. The plot shows the number of misclassifications as a function of number of neighbors evaluated (see Figure 2.27).

Figure 2.27
Total misses object



This is a pretty straightforward classification problem so we see that there are no mistakes in classification until 9 neighbors are used. Pirouette sets the optimal number of neighbors (the diamond icon on the Total Misses plot) to that value of k that produces the fewest mistakes. However, it might be risky to use only one neighbor with real data; its is often suggested to use at least 4 or 5 neighbors if that produces an acceptable success rate.

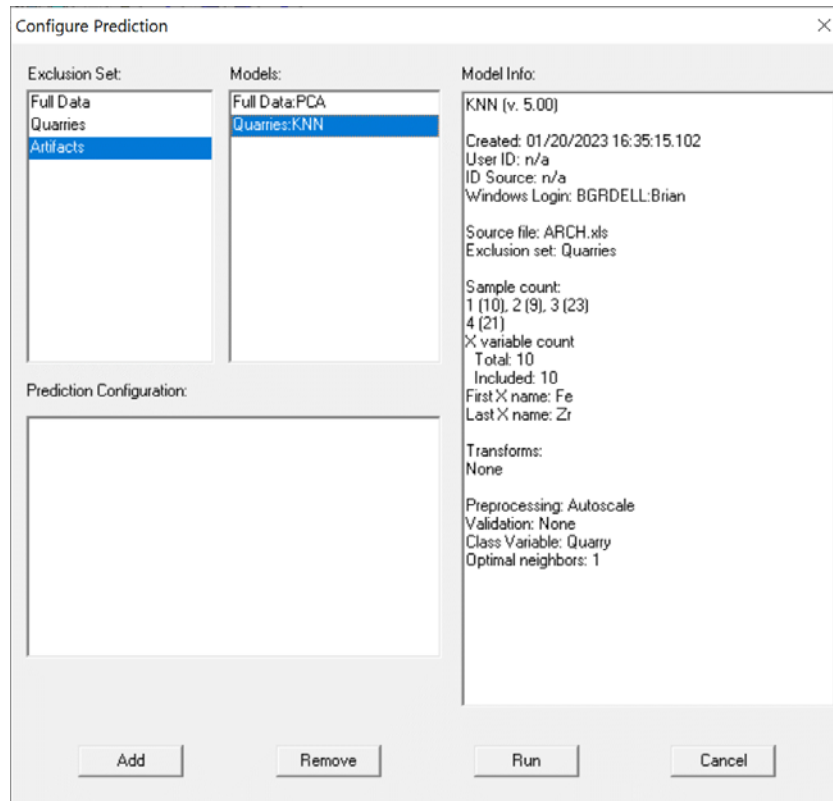
- Click on the Total Misses trace at the vertex where K = 4

Now, we are ready to do a classification on our unknown data, the artifacts.

- Select Process > Predict

This presents a Configure Prediction dialog similar to that for configuring a run (see Figure).

Figure 2.28
Configuring
Prediction



- Choose Artifacts from the Exclusion Sets list
- Choose Quarries:KNN from the Models list
- Click on Run

A new folder under the Artifacts icon in the Object Manager is created. Opening this folder reveals the simple set of KNN prediction results. Drag the Class Predicted object to the work area (see [Figure 2.29](#)).

Figure 2.29
Class Predicted
Object

1,1		3.000000				
		1	2	3	4	5
		Class				
1	s2112909	3.0000				
2	s3111309	1.0000				
3	s4111313	1.0000				
4	s5116953	1.0000				
5	s402	2.0000				
6	s4121313	2.0000				
7	s5121	2.0000				
8	s1132909	3.0000				
9	s2132910	3.0000				
10	s3132910	3.0000				
11	s4136953	3.0000				

This table shows the predicted category for each of the samples in the Artifacts subset. You could now run the same analysis using the SIMCA algorithm, which is a little more involved than KNN, and is an exercise left to the reader.

Review

For the ARCH data set, we find that samples of obsidian analyzed by X-ray fluorescence can be classified by applying Hierarchical Cluster Analysis and Principal Components Analysis. Collected artifacts have elemental compositions similar to three of the four quarries sampled. One quarry appears so dissimilar to the artifacts that it is not considered an artifact source. In addition, one artifact sample (#75) was identified as a possible outlier.

REFERENCES

1. Kowalski, B.R.; Schatzki, T.F. and Stross, F.H. "Classification of Archaeological Artifacts by Applying Pattern Recognition to Trace Element Data." *Anal. Chem.* (1972) 44: 2176.

Regression Tutorial

Contents

The Basics	3-1
Exploratory Data Analysis	3-7
Calibration and Model Validation	3-16
Review	3-33

This chapter is designed to introduce you to the power of multivariate analysis by working through an example problem that will result in the creation of a regression model. As in classification analysis, the early steps leading to a regression model are essentially the same: define the problem, organize the data and check their validity. Checking the appropriateness of the data is the realm of exploratory data analysis. When we are convinced that the data we have collected are acceptable for further work, the task will be to build a calibration model for the concentration or property of interest.

The example in this chapter presents regression analysis as seen through the problem of using near infrared spectroscopy to predict a physical property of gasoline (in this case, the octane rating of the gasoline). This walkthrough is organized into steps in the order that they would be approached for a typical problem. It also contains specific keystroke directions to provide you with a vehicle for learning the Pirouette interface. For additional details on interpreting results of the processing algorithms used, please refer to [Part I Introduction to Pirouette](#).

The Basics

DEFINE THE PROBLEM

The first step in any Pirouette analysis is to define clearly the problem you seek to solve. The data analysis described in this walkthrough is an example of the use of Pirouette to model and predict a physical property of a hydrocarbon mixture. In the petroleum industry, such measurements form the basis of commodity pricing, enable quality assessment of raw materials, and check the appropriateness of process parameters. A key measurement performed on gasoline is the calculation of the pump octane number. The ASTM standard for reporting this measurement is an internal combustion engine in which octane is measured by interpolating between the nearest standards above and below the unknown sample¹. The procedure is time consuming, involves expensive and maintenance-

intensive equipment, and requires skilled labor. For these reasons, the octane engine is not well suited to on-line monitoring where the demands are continuous.

Prediction of gasoline octane numbers using spectral features in the near infrared (NIR) has been studied by a number of research laboratories and has been implemented in many refineries². The data set described in this study will illustrate how to deal with similar problems in your own laboratory or process setting.

The purpose of this study is to assess whether a NIR spectroscopic technique can satisfactorily replace an octane engine for process quality control:

- Are the errors associated with the optical approach similar to errors we find in running the conventional analysis?
- Are there external sources of error that are of concern if the NIR approach is to replace or augment the ASTM standard?

ORGANIZE THE DATA

Sixty-seven unleaded gasoline samples were collected from a refinery and analyzed using a spectrophotometer. The wavelength range was 900 to 1600 nm with a 1 nm increment between data points. The data were assembled into a Pirouette file by merging the individual spectroscopy files. A summary of the results of the chemometric analysis described in this chapter has been published³.

In order to speed the analysis in this walkthrough, the original data were compressed into a smaller file by deleting all but every 20th data point for all 57 files. The resulting file, therefore, spans the same wavelength range in the near infrared but at a fraction of the resolution.

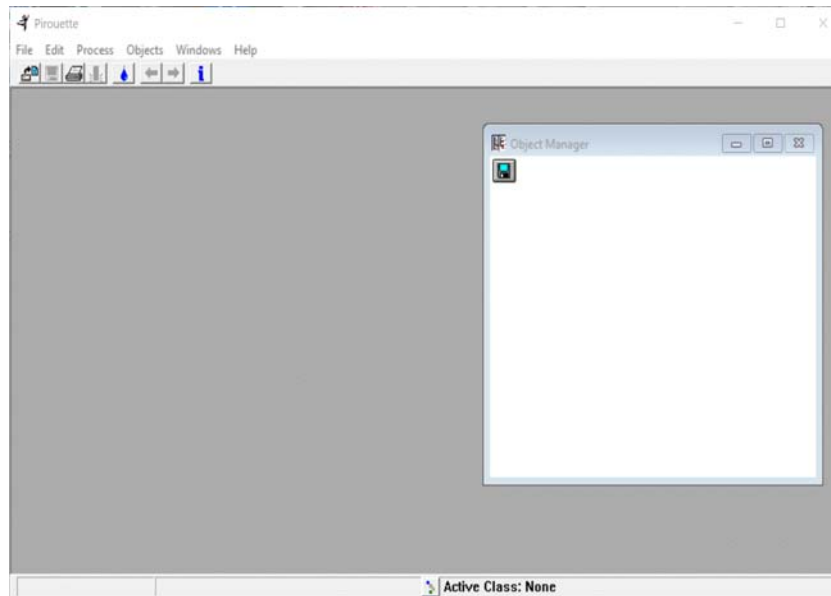
Normally, we would not throw away data prior to the analysis, but experience with examples of applications in the NIR shows that good models can be created even with very low resolution data (see the discussion in Kelley et al.²). The model we build will not be as good as the one we could make with the full 1 nm data set, but for the purpose of the walkthrough, the 20-fold increase in speed makes the analysis instantaneous.

READ THE FILE

Start the program by clicking Start > Programs > Infometrix > Pirouette > Pirouette 5.0. Pirouette will load and the screen will appear as shown in the figure below. Before proceeding, go to Windows > Preferences > Chart > Window Attributes and change the Maximum Number of Windows Created value to 2. Click on OK to close the dialog box.

Note: *You should change this value back to 0, the default, after the exercise.*

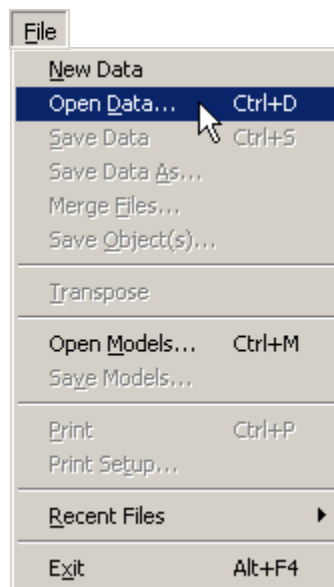
Figure 3.1
The Pirouette start up screen



Next read in the octane data set:

- Click on File from the menu bar
- Click on the Open Data option as shown in [Figure 3.2](#)

Figure 3.2
Selecting Open Data from the File menu



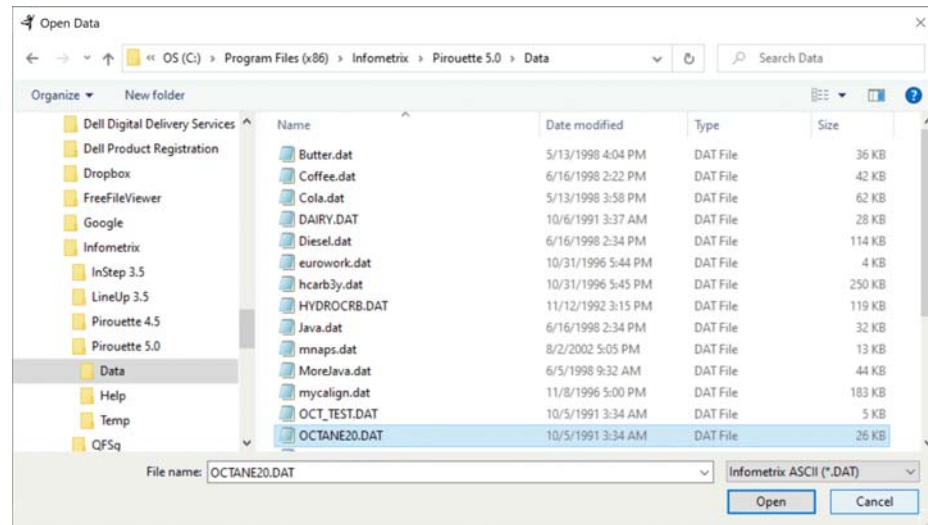
Note: An alternative is to click on the Open File button in the tool ribbon, just below the File command.



The dialog box that comes to the screen allows you to navigate through the subdirectory structure of your hard disk drive to find files. You can access any disk drive or subdirectory configured in your computer system. The layout of the Open Data dialog box is shown below.

3 Regression Tutorial: The Basics

Figure 3.3
The Open Data
dialog box



The Open Data dialog box lets the user change the directory and filter, by file type, the files that come to view in the list box. The data set OCTANE20.DAT is found among the list of ASCII files in the DATA subdirectory. To see a list of all files in the DATA subdirectory that have the DAT extension,

- Browse to the Data directory in the Infometrix\Pirouette folder
- Click on the drop down list on the lower right
- Choose the Infometrix ASCII (*.DAT) selection by clicking on that entry

To open OCTANE20.DAT,

- Click on it in the Files list box
- Click on the Open button

and the data file is read into the computer memory and made available for processing. The name of the file will now appear at the top of the screen and in the left side of the Object Manager. Click on the data icon named Full Data, below the file icon, and drag it over any blank space in the Pirouette workspace, then let go; a window appears showing a table of values.

Figure 3.4
The OCTANE20 data
set

1,1		0.214597			
		1	2	3	
		900	920	940	960
1	S001	0.214597	0.236415	0.211047	0.19793
2	S002	0.219759	0.240202	0.216255	0.20359
3	S003	0.216448	0.237912	0.212937	0.19998
4	S004	0.220137	0.240330	0.215534	0.2037
5	S005	0.213751	0.236162	0.211233	0.1972
6	S006	0.214270	0.236772	0.211726	0.19776
7	S007	0.220322	0.240446	0.215729	0.20388
8	S008	0.216660	0.238323	0.212925	0.19997
9	S009	0.213867	0.236291	0.211363	0.19756
10	S010	0.221843	0.241514	0.216588	0.20536
11	S011	0.215078	0.236834	0.211281	0.19822
12	S015	0.221395	0.241270	0.217794	0.2056

EXAMINE THE DATA

The Pirouette spreadsheet organizes data with samples presented as rows. Each sample's name is shown as the row label (for example, the name of the first sample is S001). Measured variables are displayed as columns, and the variable names are presented as the column titles (for example, 900 is the name of the first variable). You can navigate through the spreadsheet using the scroll arrows to see the absorbance data for the different samples and variables (wavelengths). Scrolling to the extreme right side of the spreadsheet also shows the single dependent variable, labeled Octane.

Note: In most spectroscopy files, you will find it faster to jump to the dependent variable column rather than use the mouse to reposition the elevator box (or scroll). To jump to the dependent variable column, press the Y button on the Pirouette ribbon.



We can easily convert the spreadsheet view to show the spectra graphically by clicking on the line plot icon. This converts the spreadsheet into a spectral view as shown below.


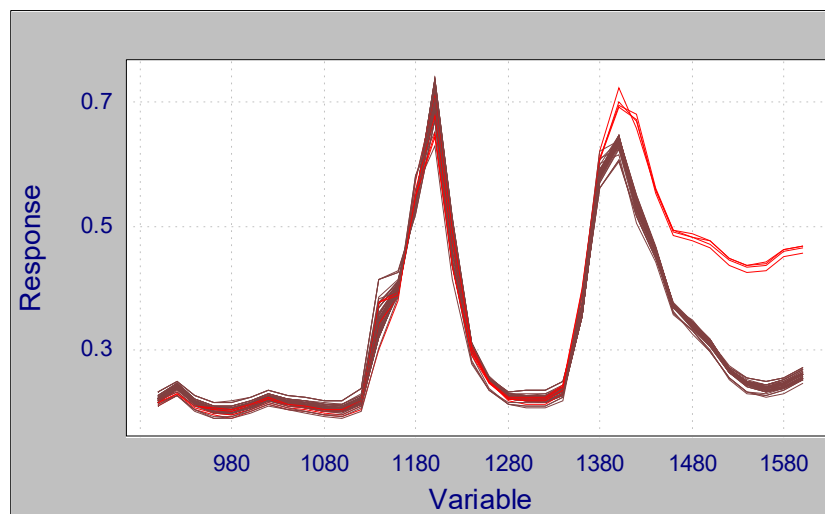
- Click on the Line Plot button 

Figure 3.5
Spectral view of the
OCTANE20 samples



The line plot that is produced is the traditional spectral view of the data. The only difference is that the x-axis is labeled with the generic name “Variable”. To label this axis for the application,

- Select the Windows > Preferences > Chart > Label Attributes menu item
- Change the Variable Axis Label to “Wavelength”
- Close the Full Data view you have open and drag a new copy of Full Data
- Switch to a line plot to display the plot with the new x-axis label

Simply by plotting the data, it is clear that there are two distinct groups present. Four of the spectra show higher absorbances in the range of 1400 to 1600 nm. This will be further elucidated in the [“Exploratory Data Analysis”](#) on page 3-7.

- Move the mouse cursor to the ribbon and select the Identification tool

The mouse cursor appears in the plot as a question mark and allows you to identify any of the lines in the graph by positioning the dot of the cursor over the position of interest and clicking the mouse button. The line identification will appear as long as the mouse button remains depressed, and the information will update as you move the cursor. As an option, you can investigate the line plot in further detail using the magnifying glass cursor. To access this enlarging tool:

- Move the mouse cursor to the ribbon and select the magnifier tool
- Click-drag to draw a rectangle on screen
- Release the mouse button to magnify the area
- Click in the plot area using the right mouse button to return the plot to its original scale

The default display is with all samples shown in the plot. This allows us to ensure that the data is of reasonable integrity and that there are no obvious problems before we start. Using the Selector Tool, we can remove some, all, or specifically select individual spectral samples for viewing. Selecting this tool will bring a context sensitive dialog box to the screen.


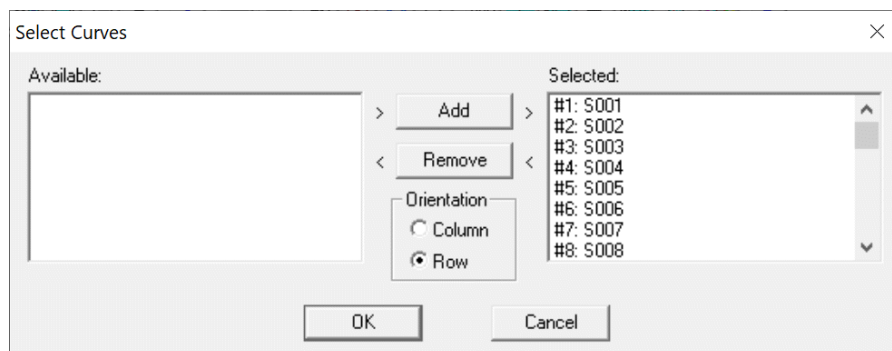
- Click on the Selector button in the ribbon 

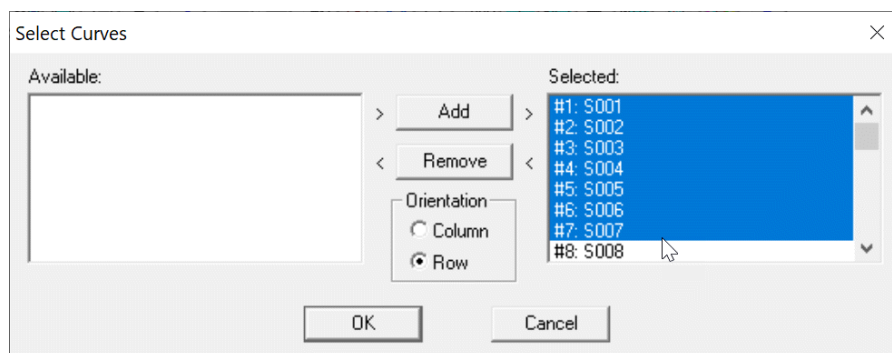
Figure 3.6
The Selector dialog box



The scrolling list on the left contains the sample names for all samples not currently plotted (none are listed); the list on the right shows those currently in the plot, which now includes the names of all the samples. Removing and then re-adding samples to the view is easy: simply double-click on a sample name or highlight a sample (or samples) and click on the Remove button. To select a number of samples at once:

- Click on the first sample in the list of available samples to highlight it
- Scroll to any position of the desired list by moving the elevator bar at the right side of the list
- Hold the Shift key down and click on the last sample name to be moved (as demonstrated below)
- Click on the Remove button to move the affected spectra
- Click on the OK button to exit the dialog box and plot the data

Figure 3.7
Removing selected spectra from the plot view



Exploratory Data Analysis

The first step in the multivariate analysis of our data set is to employ two general pattern recognition algorithms to explore the data. These techniques provide information that extends beyond simple examination of the data as line plots or selected 2-D and 3-D scatter plots.

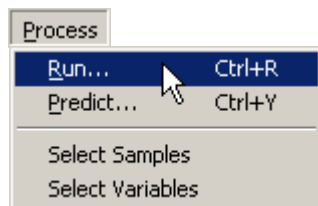
SET UP

Exploratory data analysis primarily consists of Hierarchical Cluster Analysis and Principal Component (Factor) Analysis methods and examination of the graphics that these methods provide for suggestions of relationships between variables, between samples

and possible outliers. Prior to performing the exploratory data analysis, we need to set any data processing options that we want applied to the data (for a discussion of these options, see “Preprocessing” on page 4-26). Preprocessing options are set for the algorithms in the Configure Run dialog box.

- Click on the Process menu
- Click on the Run menu item as shown in Figure 3.8.

Figure 3.8
Selecting Run from
the Process menu



The dialog box for selecting the data processing options and their control configurations will be displayed. This will allow you to customize the processing parameters for the exploration algorithms, HCA and PCA.

When the dialog box appears on screen, the HCA algorithm (upper left list) is initially highlighted, and the available options are presented in a dialog window to the right. NIR spectral data should be centered prior to analysis. To do so,

- Position the mouse over the drop down list to the right of “Preprocessing”
- Click on Mean Center to highlight and select that option (it may already be selected)

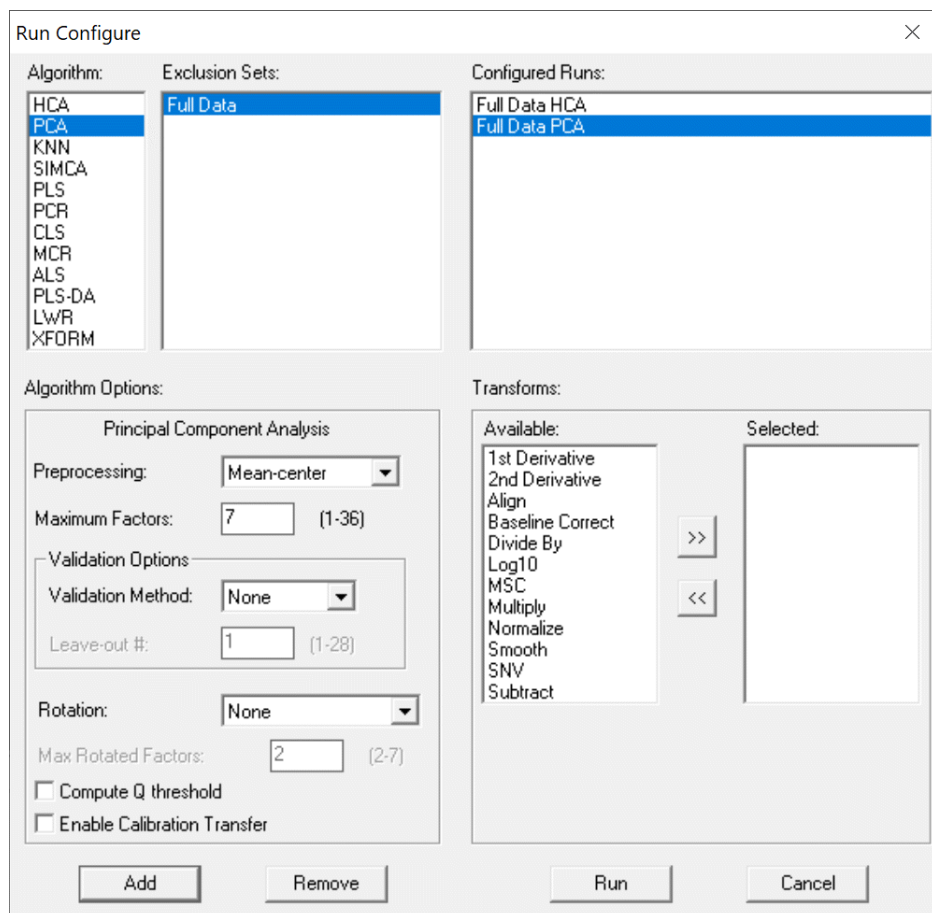
Mean Center will now appear in the box, indicating the selected preprocessing option. The Distance Metric option listed is Euclidean. There are seven alternative clustering techniques available (for additional detail, please see “Linkage Method Definitions” on page 5-2). For this exercise, we want to use the Group Average method.

- Position the mouse over the drop down list to the right of “Linkage Method” and click
- Drag the mouse so that Group Average is highlighted and release

With all HCA processing options now selected,

- Click on the Add button in the lower left corner of the Configure Run dialog window and this algorithm is added to the sequence order of methods for data analysis. Repeat this process for Principal Component Analysis (PCA) so that the processing will be consistent and allow comparisons:
 - Click on the PCA algorithm in the upper left of the dialog box to bring the PCA options to the display
 - Select Mean Center as the preprocessing option, as done for HCA
 - Set the Maximum factors to 7
 - Click on Add to add PCA to the Run with selected and default options as shown below

Figure 3.9
Adding PCA to the
run configuration



HCA and PCA are now set up to be run on mean centered data. These settings will remain active until they are changed or until a new data file is loaded.

Note: *If you also wanted to run a different clustering technique (say Flexible Link), you could change the clustering option in the Options area to Flexible Link after rehighlighting HCA in Algorithms, add this method/option to the Run Configuration List, and you would have both clustering results when complete.*

RUNNING THE EXPLORATION ALGORITHMS

With the two algorithms added to the Run Configuration with the chosen options, all that is required to initiate processing is to click on the Run button at the bottom of the window.

- Move the cursor to the Run button
- Click with the mouse to start processing all algorithms added previously

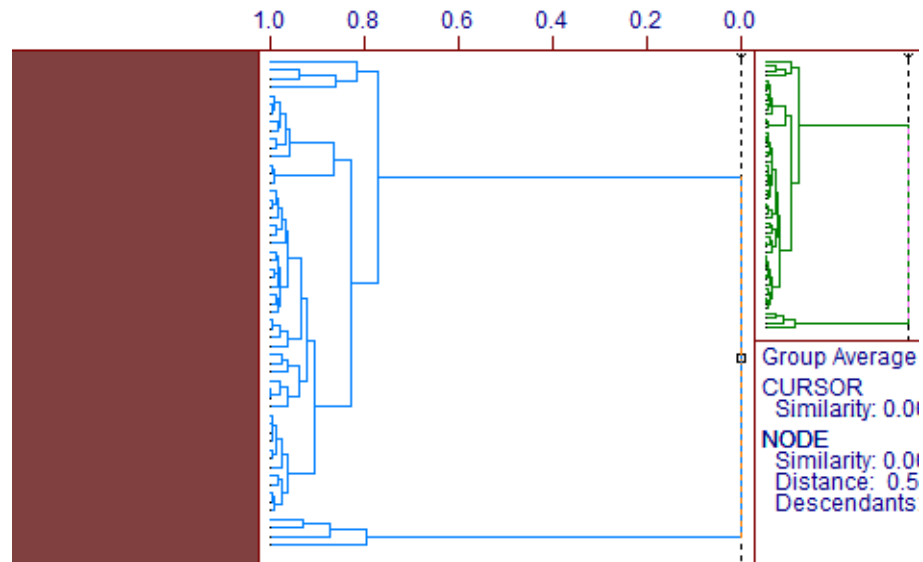
During the processing time, a Run Status message window opens up and displays the progress while the program is working. This window allows you to abort after a run if necessary. Upon completion of the processing of an algorithm, this status screen will change to indicate that the processing is complete. At the end of all runs, the status window will close (unless a run problem was encountered).

3 Regression Tutorial: Exploratory Data Analysis

The calculated objects for the selected algorithms are displayed in chart windows stacked on top of the original Full Data spreadsheet display (if your Maximum Windows preference is set to zero, drag out the HCA results to form a new chart window). The individual algorithm results can be viewed in turn or rearranged in the display area to examine a variety of results and different views. We will first examine the HCA results.

- Click on the Full Data:HCA name in the title bar at the top of its window to bring the display to the foreground
- Double-click on the dendrogram plot to bring its subplot to fill the window

Figure 3.10
HCA results



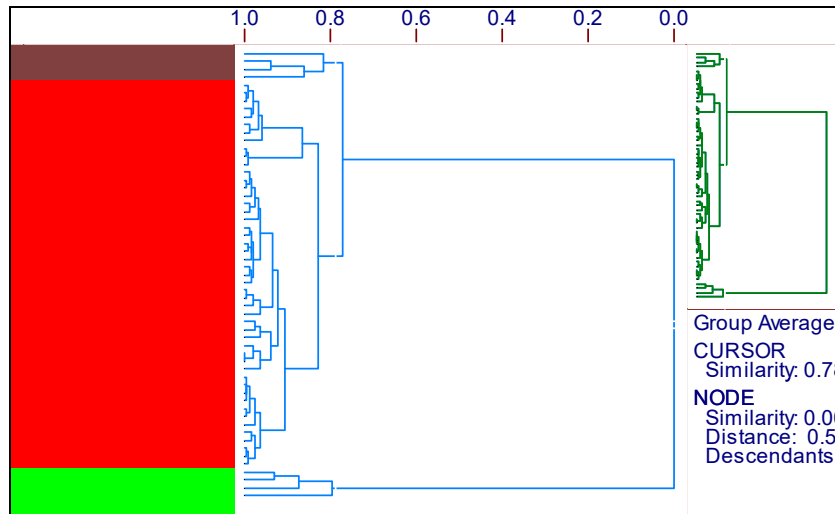
The HCA results are presented in the form of a plot called a dendrogram. This display organizes samples in the data set based on their multivariate similarity (for details, please see “Distance Measures” on page 5-2). This graphic is presented with the pointer tool activated. We will use this pointer to set the similarity cursor and explore some of the details of the dendrogram.

- Position the cursor over the vertical dashed line near the inverted caret (or v), under the 0.0 mark, where it turns into a double arrow.
- Click-drag on the line to the left until a similarity value of about 0.78 is achieved
- Release the mouse button

The color bar display on the left of the dendrogram changes into separate colors that distinguish samples grouped on the same branches cut by the similarity cursor (see below). This position has cut three branches and we can begin to explore the membership on each.

Note: *At any time, we can reposition the similarity cursor at a different value to explore other groupings that result. If a particular grouping is of sufficient interest, the assignments can be saved to the main data file by selecting the Edit menu and the Activate Class menu item or by hitting the key combination Ctrl + K.*

Figure 3.11
Setting the similarity



The dendrogram gives information about individual gasoline samples in the data set and their similarities to other samples that may suggest natural groupings. It is advantageous to compare the dendrogram clusters with the PCA scores plots to examine the extent to which apparent groupings are exhibited in both views.

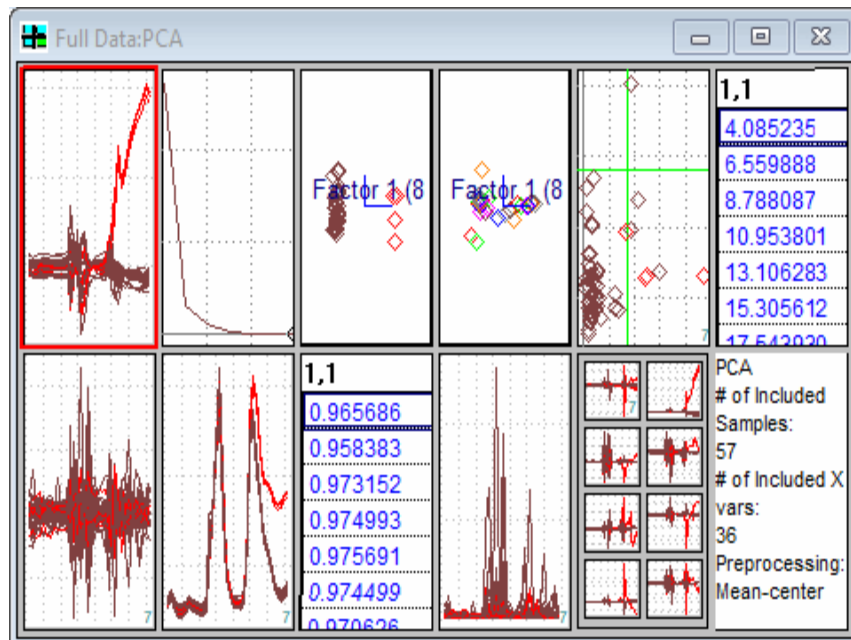
DATA INTERPRETATION

For the next few minutes, let's work with the data and the displays to achieve a better understanding of how data can be handled within Pirouette. First, it is important to note that all of the data displays in the program are linked such that interaction with the data displayed in one window may affect other windows.

- Click on the window showing Full Data: PCA to bring it to the foreground

Twelve objects comprise the PCA window as shown below.

Figure 3.12 Display of PCA results



3 Regression Tutorial: Exploratory Data Analysis

The collection of results from PCA include a multivariate description of the data matrix in the form of scores and loadings and associated variances, as well as a variety of diagnostics useful for evaluating the analysis. We will initially examine the interactive linkage between HCA and PCA scores views to illustrate the more distinctive samples.



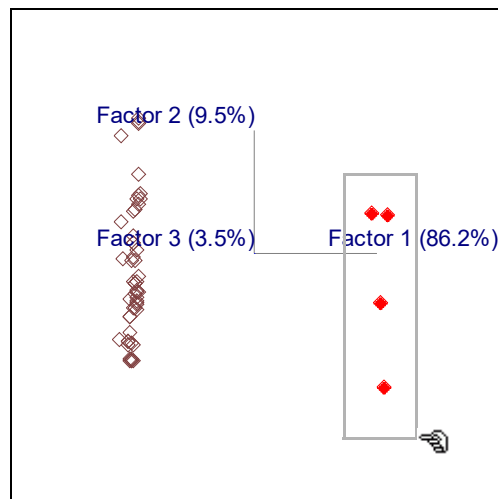
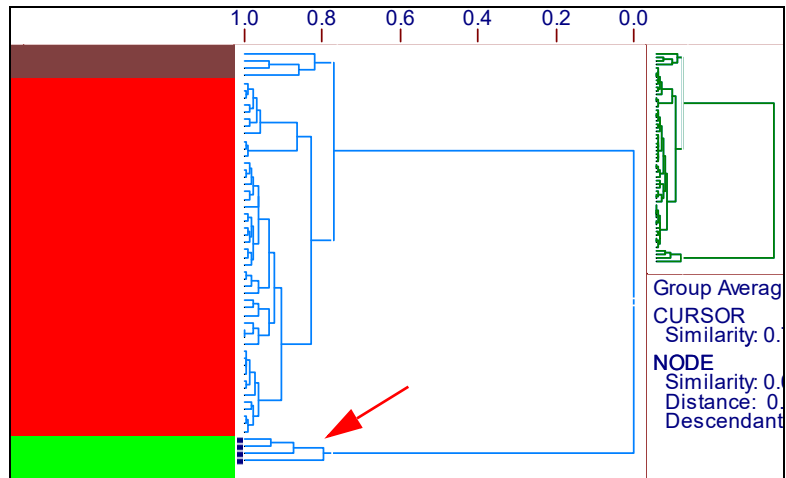
- Click the mouse on the scores plot (the third subplot) to make it active
- Click on the Zoom button in the tool bar 
- Click on the Pointer button 
- Click and drag the mouse within the scores plot to create a box around the four data points that are separate from the bulk of the samples
- Release the mouse button to highlight the aberrant points, as shown next

Figure 3.13
Click-drag-release to
select contiguous
points



Immediately upon release of the mouse button, the samples highlighted in the scores plot will also be highlighted in all other plots containing sample information (in this case the dendrogram). This effect is noted in the following figure. Click on the dendrogram window to bring it to the foreground. Note that points selected in the scores plot are highlighted as well in the dendrogram and occur together on the lowest major branch, highly dissimilar from all others.

Figure 3.14
Points highlighted in the dendrogram



The scores plot is a 3D mapping of the sample data. To take advantage of this feature, we will employ the Spinner tool.


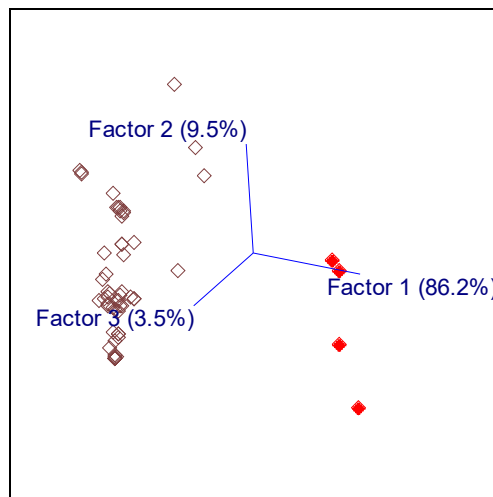
- Click on the PCA window to make the scores window active
- Move the cursor to the ribbon and click on the Spinner button 
- Return the cursor to the scores plot and click-drag the Spinner across the plot from right to left to rotate the data points, as shown here

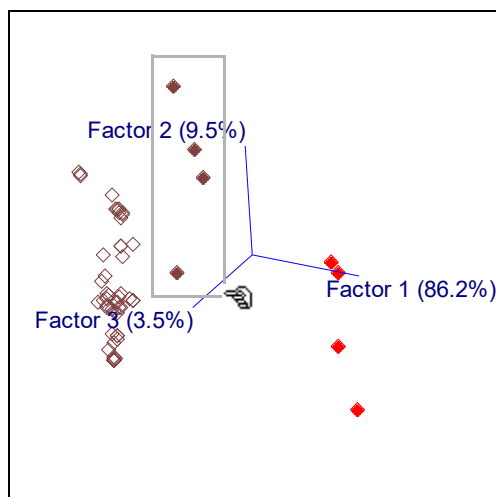
Figure 3.15
Rotating the scores gains a different perspective



The first set of four outliers in the data were identified immediately upon display of either the dendrogram or the scores plot. By rotating the plot, we can see a second set of possible outliers separate from the majority of the samples. To highlight these samples in addition to those highlighted previously,

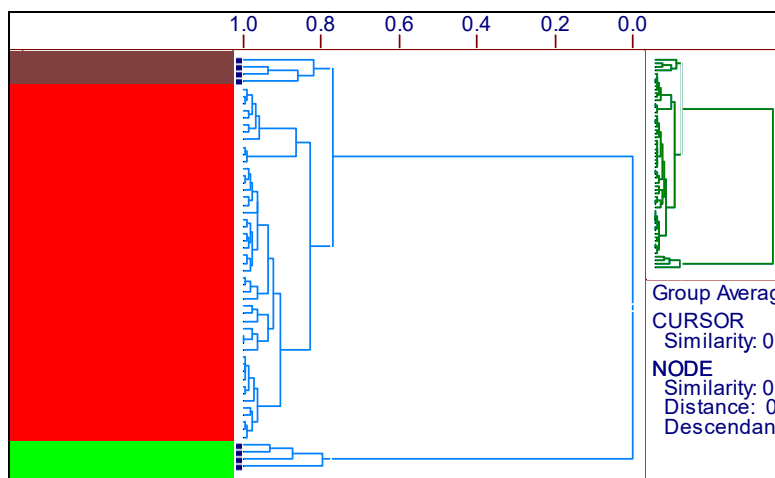
- Click on the Pointer button in the ribbon
- Move the cursor back over to the scores plot
- Hold the Control key down as you click-drag forming a rectangle around this second set of (maybe aberrant?) points as demonstrated next

Figure 3.16
Highlighting a
second set of
possible outliers



The linking between the scores view and the dendrogram confirms the status of these eight samples as belonging to two separate groups of possible outliers; the samples can be seen as distinct from the main body of data as well as being distinct from one another. This is illustrated in the next figure.

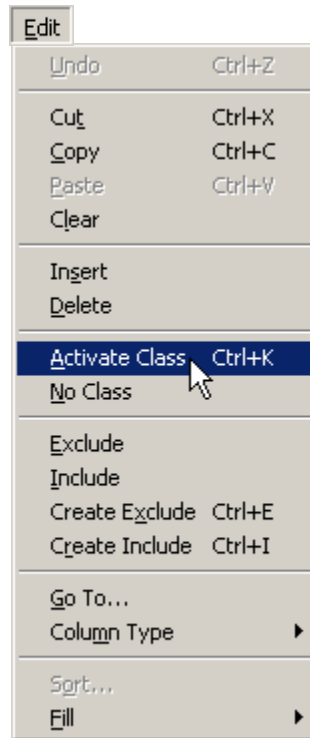
Figure 3.17
Linking between
scores plot and
dendrogram view



The dendrogram can also be manipulated to emphasize outlier samples. Because we had previously set the similarity cursor at 0.78, we have defined several potential groupings and individual distinctive samples. The dashed line cuts 3 branches. One set of four samples (at the bottom of the displayed dendrogram) is most unique and represents the spectra we had noted earlier as unusual (see [Figure 3.5](#)). A second set of four samples at the top of the dendrogram are more similar to the bulk of the gasoline samples in this study, but are still somewhat distinct from the majority. These three subdivisions of the data are indicated by different colors in the color bar at the left side of the dendrogram display (created when we moved the similarity cursor in [Figure 3.11](#)). We can use this subdivision to color identify the samples on the scores plot and add an identifying class column in the data table, by using the Activate Class feature.

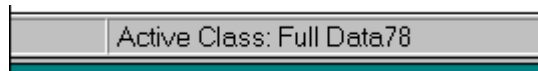
- Select the Activate Class option from the Edit menu (or use the Ctrl-K keyboard equivalent)

Figure 3.18
Activating a class to
map dendrogram
colors to other views



With this action, a new class variable has been incorporated into the data file that will remain with the file when saved. The second effect is that the activation of a class variable from the dendrogram view colors the data points in all other views. Pirouette flags the name of the active class in the message area at the bottom of the Pirouette window as.

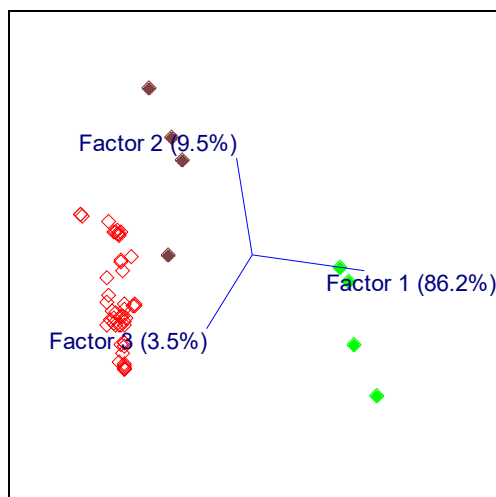
Figure 3.19
Message area
showing the name of
the active class



The default name of the class created from the dendrogram view is established by merging the name of the data object that was used to create the dendrogram with the similarity value marked by the cursor position. In this case, the dendrogram was built using the “Full Data”, and the position of the similarity line would round to 0.78. Thus, the default class name is “Full Data78”; the name can be modified by editing in a spreadsheet view of the data. For additional detail, please see [“Creating Class Variables”](#) on page 12-27.

As mentioned above, the action of activating a class from the dendrogram view also maps the colors defined by the dendrogram into all other views. Because the scores plot is sample-oriented, the active class will repaint the colors in the scores view to correspond as shown below.

Figure 3.20
PCA scores
recolored to reflect
the active class



THE NEXT STEP: CALIBRATION

The exploratory data analysis has shown us that there is structure to the octane data and that there appear to be three groupings in the collection of 57 samples. The reason for concern about the clustering of the octane data is the following question:

Do these clusters represent different populations of gasoline samples or do they indicate that this training set is not fully representative of all possible samples?

In either case, by retaining all samples during calibration, the final model will contain errors that relate to this inhomogeneity. The ideal case is to have a scores plot that shows a homogeneous cloud of points and a dendrogram that shows very little cluster tendency. As it turned out, when this data set was originally analyzed, we contacted the refinery and found that the four most obvious outlying gasoline samples contained alcohol, while the four less obvious outliers contained MTBE (ether) as octane boosting agents. These oxygenates were present at a level of roughly 10% by volume.

If there were enough samples in each of the two unusual groups (we would like to see about a dozen or so samples in each category in order to be assured of reasonable statistical accuracy), the course of action would be to run a classification analysis (KNN and/or SIMCA) and then develop separate calibration models for each of the three groups.

We will retain the outlying gasoline samples for the first pass of calibration in order to demonstrate how outliers are detected in PCR and PLS. To create the most representative model, the outliers will then be removed and the resulting model will be tested on new data.

Calibration and Model Validation

To process the data, we can build either Principal Component Regression (PCR) or Partial Least Squares (PLS) regression models of the data, or both. We could compare the results of the two modeling techniques and choose to save one or both as a model for future prediction use. Several interpretation concepts will be introduced which are not completely described here, in order to keep this walkthrough reasonably brief. As questions arise, please refer to [Chapter 7, Regression Methods](#).

SET UP

The first task is to set the processing techniques to be used for the analysis. To be consistent with the exploratory analysis, we will choose to mean center the data:

- Click on Process in the menu bar
- Click on the Run item in the Process menu
- When the dialog box appears on the screen, click on the PLS Algorithm
- Choose the preprocessing option Mean-center
- Click on Add to include PLS in the Run Configuration

Repeat this selection process for PCR so that the processing will be consistent between the two methods and allow comparisons:

- Click on the PCR Algorithm
- Select Mean-center
- Click on Add

The PLS and PCR calibration algorithms are now set up to be run after first mean centering the data.

CALIBRATION WITH PCR AND PLS

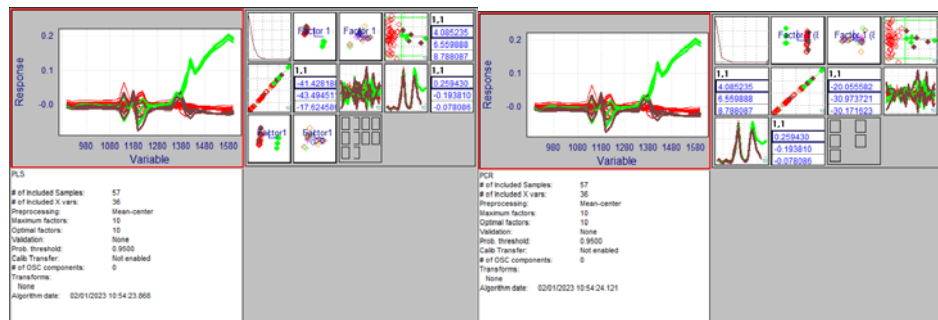
At this point, no further options need to be set. Close all windows except the Object Manager. Then, to perform a calibration,

- Click on the Run button 

The algorithms are processed in the order they are entered in the Run Configuration dialog box; in this case PLS is processed first, followed by PCR. If necessary, drag the PLS results icon, then the PCR results icon, to the desktop so that all of their results are displayed (and close any other results windows that may still be open). In [Figure 3.21](#), we have placed the PLS and PCR results side-by-side using the steps as follows:

- Minimize the Object Manager window
- Double click on the center subplot of each window to zoom the main results
- Select the Tile option from the Windows menu

Figure 3.21
The calibration results screen, for PLS and PCR



Each of the two results windows contain three sections. On the top left, you find a plot of the data after preprocessing; in this case we specified mean centering, so the data has the average values at each wavelength subtracted to highlight the spectral differences. Below this plot is a notes field that summarizes the processing performed. The multiplot shown on the right contains all of the objects computed during the processing of each al-

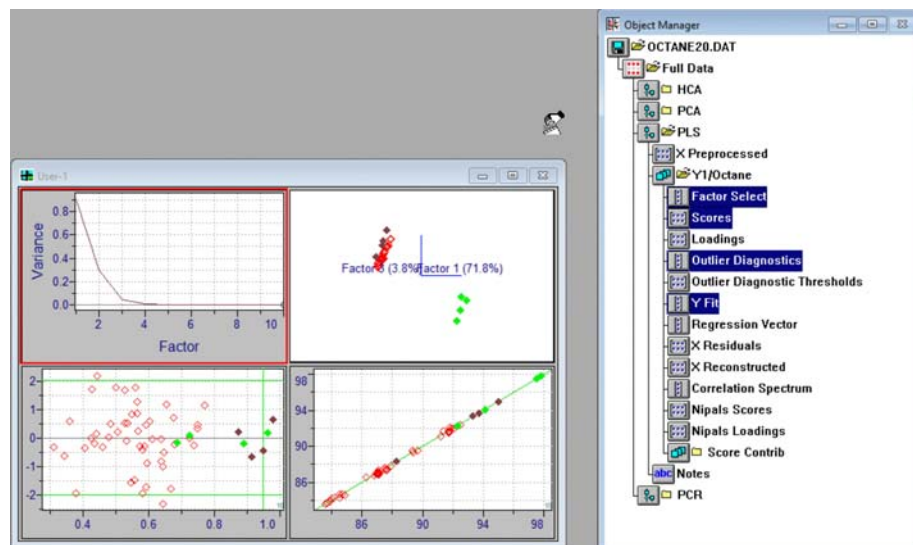
gorithm. The PCR algorithm generates eleven objects and PLS builds thirteen. These objects are used to interpret the quality of the models for doing routine prediction. For the purposes of this walkthrough, we will describe the evaluation of the data based on four PLS objects (Figure 3.22). Details of all objects generated during regression can be found in Chapter 7, Regression Methods.

To prepare a window with the four results desired,

- Windows > Close All Windows
- In the Object Manager, double-click on the PLS folder
- Double-click on the Y1/Octane folder
- Click on the Factor Select item
- With the Control key held down, click on the Score, Outlier Diagnostics and Y Fit items
- Position the cursor over one of the selected items, then drag to a clear space in the Pirouette workspace

A User window will be created like that in the following figure.

Figure 3.22
Four PLS objects

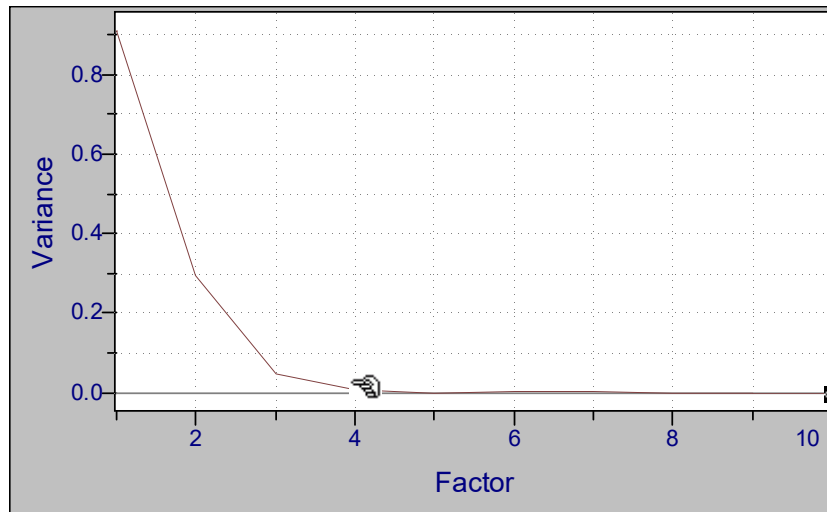


By running PLS (and PCR), we have performed a calibration, and a model has been automatically created. Models are managed as a part of the Pirouette file and are also able to be saved as separate files in either binary or ASCII form (see “Saving the Model”).

DATA INTERPRETATION

The factor select plot in the NW window (Figure 3.22 and enlarged in Figure 3.23 below) shows the variance dropping significantly as more principal components are included in the regression. Most of the model improvement occurs with the inclusion of four principal components, where the cursor is positioned in the figure, although note that the Pirouette algorithm makes a first suggestion of using ten components (as marked by the open circle on the line plot). This suggests that outliers are a problem.

Figure 3.23
The variance (eigenvalue) plot



The scores plot shows a distribution similar to the one found during the exploratory analysis phase (PLS scores do differ from the PCA scores; see “Scores” in Chapter 5). Let’s take a closer look at these outliers by rotating the scores plot:



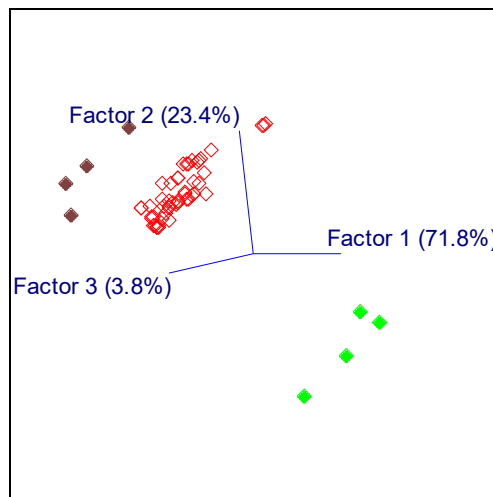
- Click on the unzoom button in the ribbon
- Double-click on the scores plot to zoom it to full window status
- Select the Spinner tool from the Pirouette ribbon to view the outliers 
- Rotate the scores to accentuate the differences between outliers and non-outliers
- Turn off the labels by clicking the Label button 

Figure 3.24
PLS scores



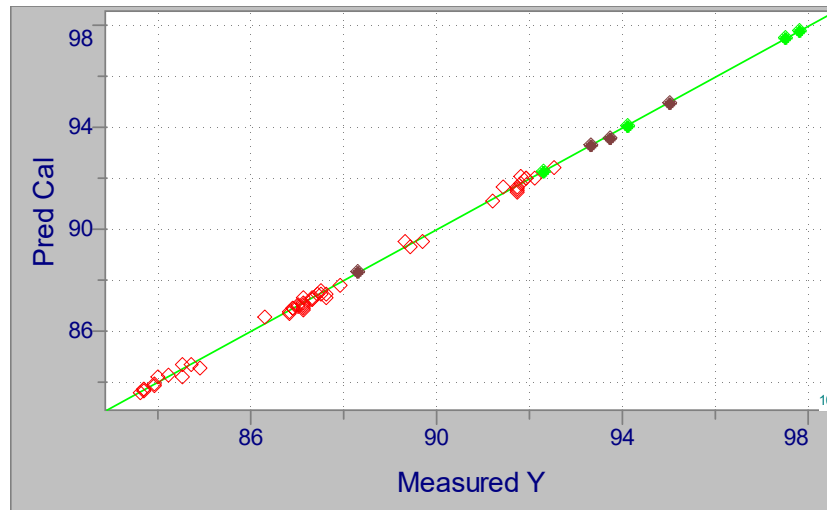
Note the coloring of the points is retained from the activation of a class variable (refer to Figure 3.18).

Despite the inclusion of obvious outliers, the model predictions fit closely to the actual octane values. The quality of the regression fit can be seen by comparing the actual octane values with the PLS fitted values.

- Click on the unzoom button in the ribbon

- Double-click on the Y Fit plot to zoom it to full window status

Figure 3.25
The fit of predicted versus actual octane values

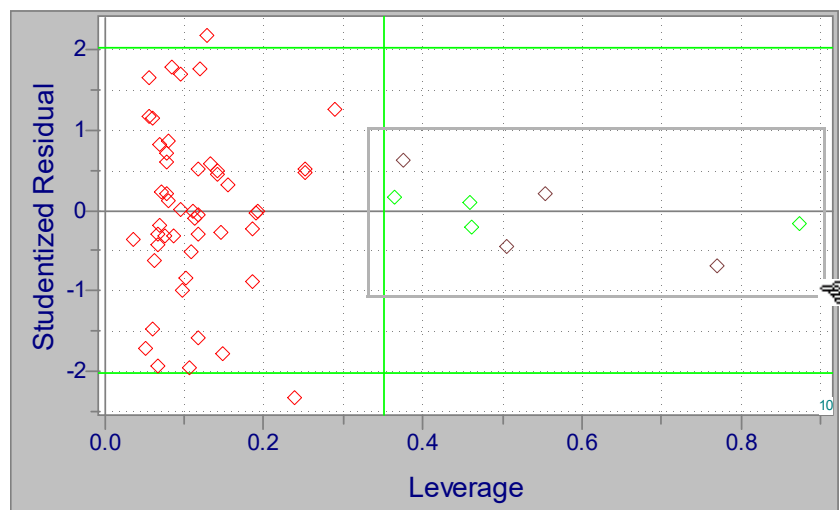


We had the suggestion, both in the error plot (where the number of components suggested by the algorithm to fit the data was high) and in the scores (where we observe three distinct clusters of data points), that there are some unusual samples in the data set. The studentized residuals plotted against leverage (Figure 3.26) reinforces the knowledge that the eight samples we flagged before are too unusual to be included in any final model (see “Outlier Diagnostics” in Chapter 7). Samples of high leverage have an undo influence on the equation of the regression line used to model the property of interest (in this case the octane rating).

Now that we have examined the outliers more closely, it is time to eliminate them from our model. The easiest method of selecting outliers is to select them from the residuals plot.

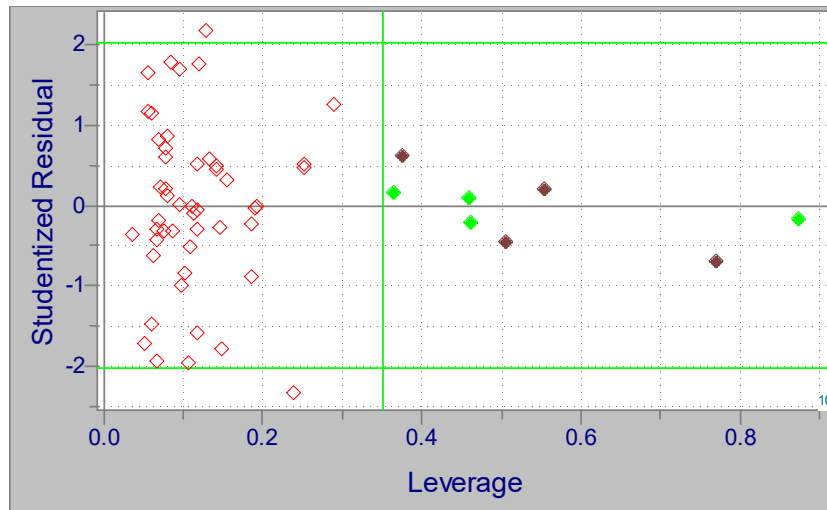
- Click on the unzoom button in the ribbon
- Double-click on the Outlier Diagnostics plot to zoom it to full window status
- Click-drag a selection box around the points that have leverage values above the cut-off (denoted by a vertical line) as in Figure 3.26

Figure 3.26
Residual versus leverage plot for identifying outliers



- Release the mouse button to leave the high leverage samples highlighted

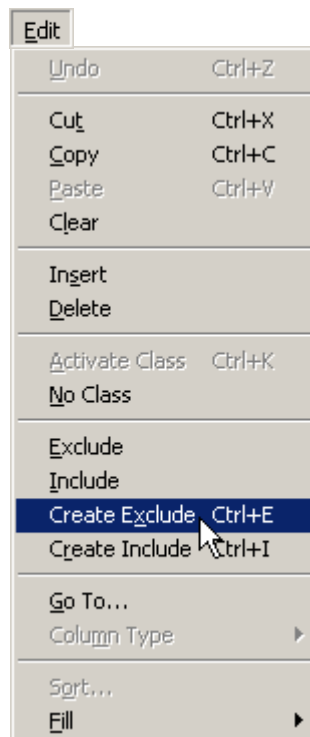
Figure 3.27
The highlighting of outliers in the leverage plot



We will now create a subset of the original data that does not include the outlying samples identified above. This subset, built by excluding some of the samples of the original is also referred to as an exclusion set.

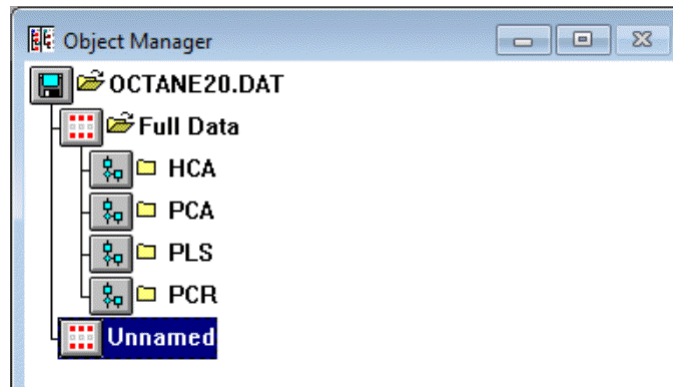
- Select the Create Exclude option under the Edit menu (or press the Ctrl-E key combination)

Figure 3.28
Creating a subset (exclusion set) of the data



A subset of the original data has now been created; this subset excludes the eight outliers. Creation of a new subset can be verified in the left side of the Object Manager, where the default subset name “Unnamed” is displayed under the Full Data icon.

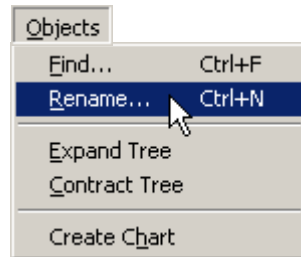
Figure 3.29
The Object Manager organizes subsets of the data file



Because Pirouette can manage a nearly unlimited number of subsets, it is useful to name the new set appropriate for the data it contains.

- Click on Unnamed in the Object Manager
- Choose the Rename option from the Objects menu (Ctrl-N)

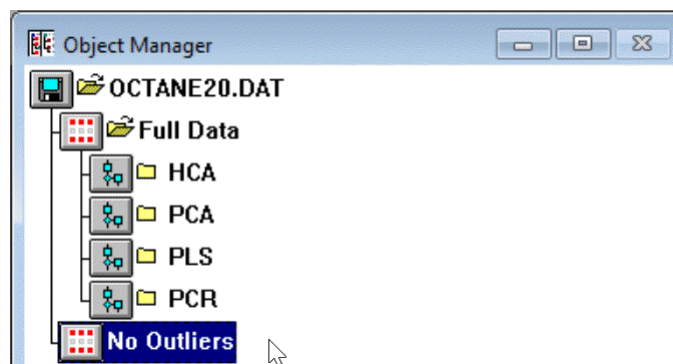
Figure 3.30
Renaming a subset




- Type No Outliers as the new name of the subset (spaces in the name are OK)
- Click on OK to establish the new subset name

You will find, on exit from the dialog box, that the subset name will take on the newly assigned name.

Figure 3.31
The Object Manager reflects all name changes



We can now utilize the No Outliers subset, without any changes in the preprocessing options for PLS and PCR, to quickly repeat the calibration.

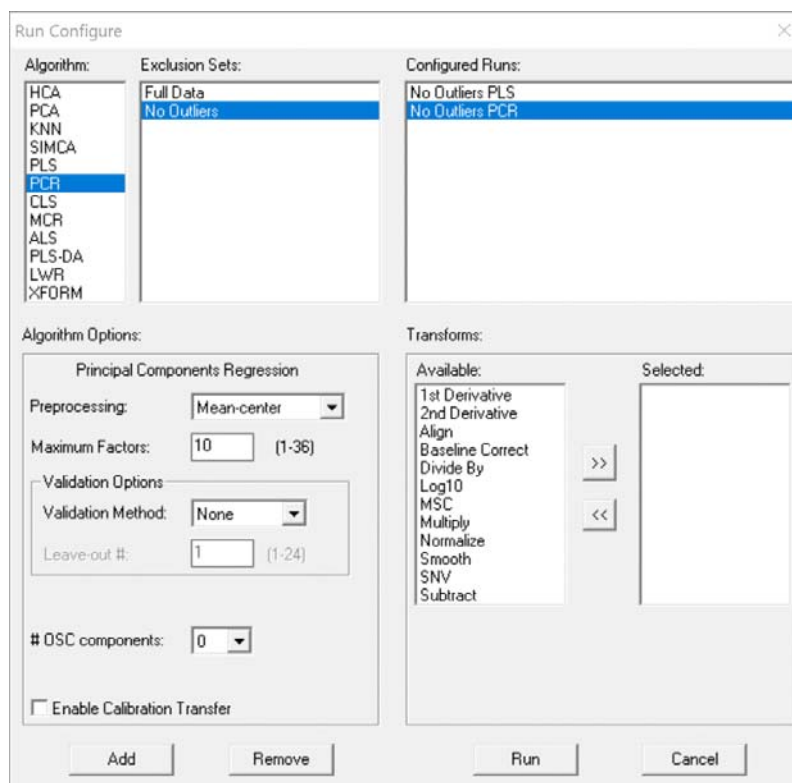
- Choose the run option by selecting from the ribbon 

and we can set up to rerun PLS and PCR using the Run/Configure dialog box. The options will still be set from the previous analysis on the full data set, so we need only:

- Select the No Outliers subset from the listing of exclusion sets available
- Click on PLS
- Click on the Add button
- Click on PCR to select that algorithm
- Click on the Add button

The run is now configured as modeled in the following figure.

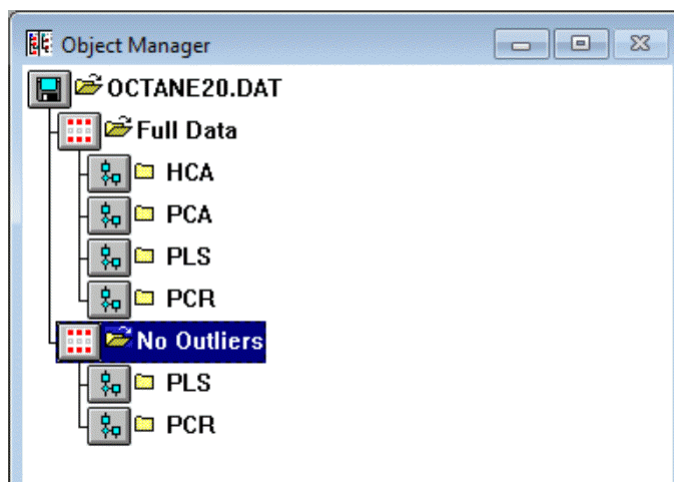
Figure 3.32
Rerunning PLS and
PCR on a subset



- Click on the Run button to start the analysis

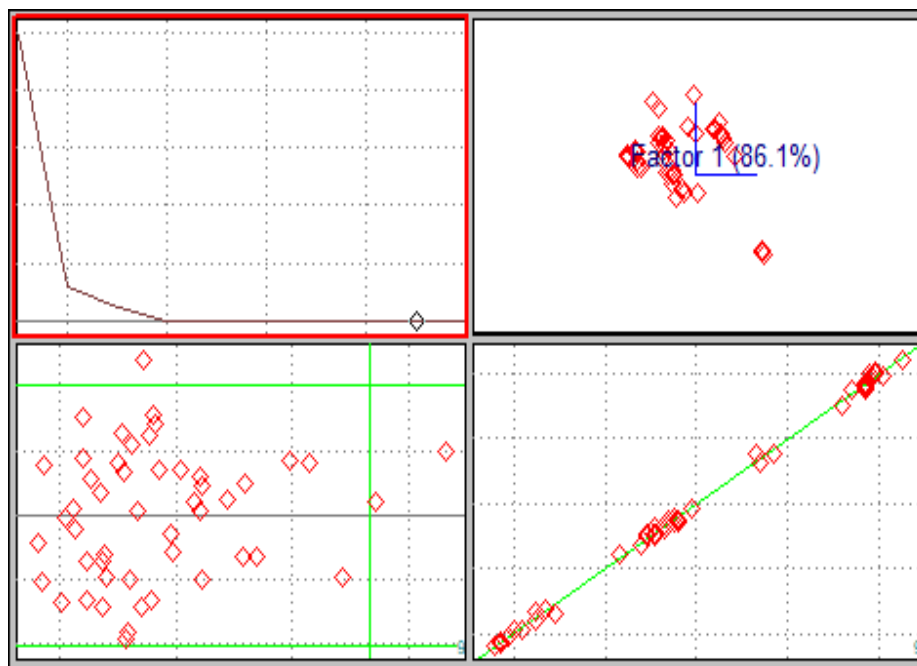
When complete, the Object Manager will reflect the newly created objects.

Figure 3.33
The Object Manager
displaying
algorithms and time
stamps



As before, PLS and PCR are processed in the order entered in the Run Configuration dialog box and a status screen is brought to the screen. When completed, each algorithm which is run can be displayed in its own window. Also, as before, we will concentrate on four of the diagnostic displays for PLS (Figure 3.34 is analogous to Figure 3.22's results for the whole data set).

Figure 3.34
PLS results with
outliers removed



The results of the analysis are significantly improved over that seen when the outlier samples were included.

Let's look more closely at the data in the NW window to get a better feel for the number of principal components necessary for modeling octane. Although Pirouette attempts to estimate the proper number of principal components by using an F test (see "Estimating the Optimal Number of Factors" in Chapter 7), the choice is not always optimal. The optimal number of PCs (principal components) to use in the model involves judgment by the user: if too few components are used, we will not have adequately modeled the avail-

able information to permit reliable future predictions; if too many components are chosen, the model will contain noise as well as information, and the predictions may be less reliable.

For example, we can change the number of principal components included in the model and see the effect on predictions and outlier diagnostics.

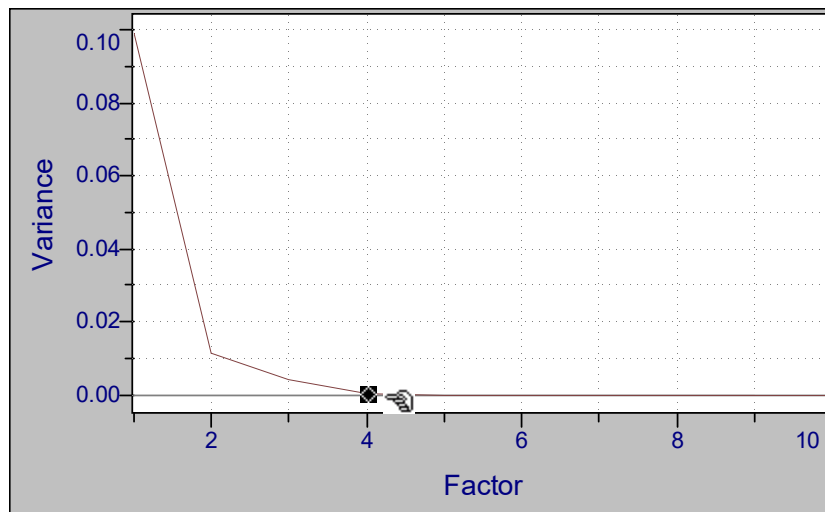
- Click-drag a second copy of the PLS Factor Select object from the Object Manager onto the Pirouette desktop

The line plot will show the diamond marker on the 9 PC position. To change the number of PCs included in the model:

- Click the mouse while positioned above a different number of PCs (as shown in the following figure)

and the diagnostics will update for every other window. You will now see the prediction plot and the residuals plot update to reflect the new level of model complexity (here at 4 components).

Figure 3.35
Setting a 4
component PLS
model



Changes to the model complexity can be done using your judgement or experience with past data sets. Better is to perform a validation of the data, outlined here and explained in more detail in [Chapter 7, Regression Methods](#).

Any model that we create is dependent on the number of principal components. By clicking on the variance plot, the two windows dependent on the number of PCs (*i.e.*, the predictions and residuals plots) will change to reflect the new model. The predicted-versus-actual plot and the residuals plot are updated. Note the number of principal components used for the model is displayed in the lower right portion of each window.

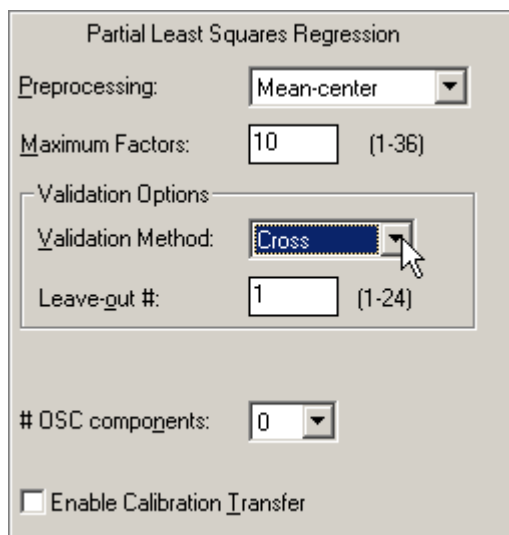
An ideal evaluation of the PLS model would be to use an independent set of data. Here, we will instead run an internal (cross) validation which will approximate results obtained with external validation.

To start a leave-one-out cross validation, we return to the Run Configuration dialog box.

- Click on the Run Configure button or select Run from the Process menu
- For the PLS algorithm, choose the No Outlier subset and set the validation options as shown here

3 Regression Tutorial: Calibration and Model Validation

Figure 3.36
Setting the options
for cross validation

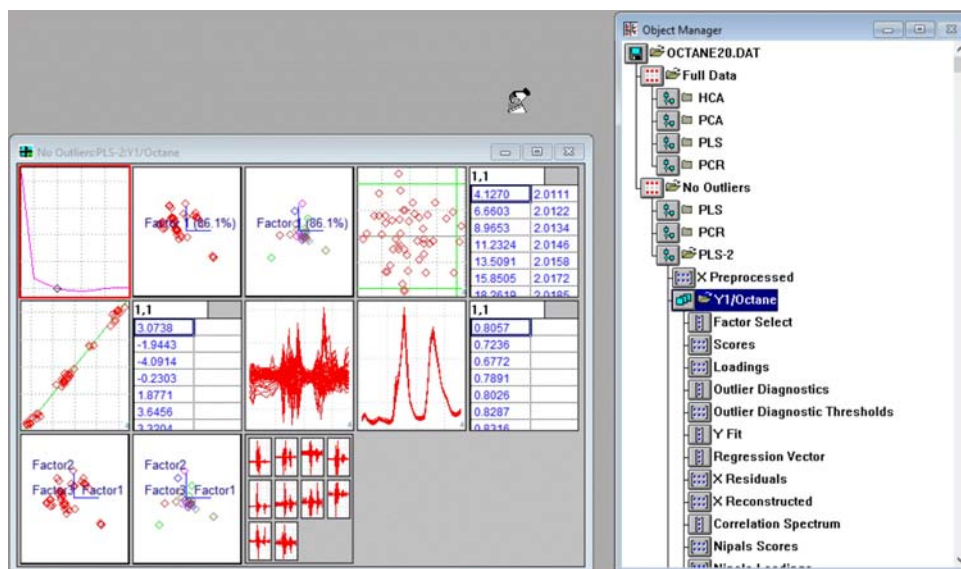


- Click on the Run button

This processing will take as long as several seconds to complete depending on the speed of your computer. Leave-one-out cross validation does what it sounds like it does; it removes one of the samples from the training set, performs a PLS regression on the remaining samples, predicts the octane value for the left-out sample and then tallies the error. Each sample is removed in this manner one time, therefore, with 49 samples in the training data, the PLS algorithm will be run 49 times in the cross validation step.

When complete, the cross validated PLS results can be displayed on screen.

Figure 3.37
Cross validated PLS
results

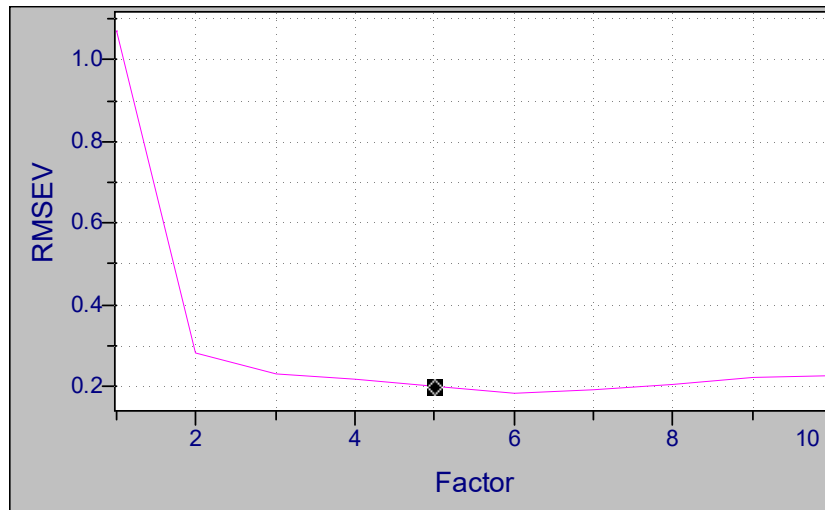


You can see that the algorithm's estimate of the best number of principal components to include in the model has changed, decreasing from 9 to 4.

- Double-click to zoom the error object (in the first subplot)

and you can see the prediction errors fall for the first 6 principal components. If more PCs are included in the model, the error starts to rise, presumably because at this point we are starting to incorporate noise in the model.

Figure 3.38
Standard error of validation



Note that the algorithm built into Pirouette chose 5 PCs, rather than the absolute minimum error at 6. The algorithm opted for fewer PCs after determining by F-test that the model improvement was not significant enough to justify the more complex model.

You can still override the Pirouette choice; the position of the diamond cursor will control the model complexity as we save the model for future work (and the next section of this walk-through).

SAVING THE MODEL

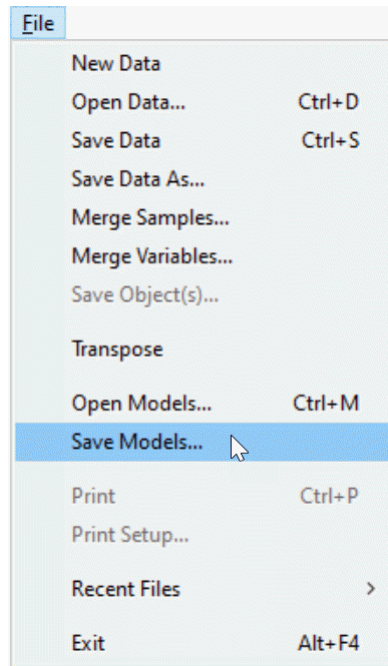
Pirouette keeps a record of all models that have been generated for a particular data file. These models are stored and managed within that file by the program. For this walk-through exercise, we have built 6 different models so far: PCA, PLS and PCR on the Full Data plus two PLS and one PCR on the No Outliers subset. Pirouette allows you to save one or more of these models as a single file.

In order to use the model we have just created, we need to save the model file as a separate file on disk. The model-save employs the what-you-see-is-what-you-get approach; the model parameters displayed on the screen at the time of the save will be the model saved: in this case, a 4-component PLS model. To save the model:

- Pull down the File menu and choose the Save Models option

3 Regression Tutorial: Calibration and Model Validation

Figure 3.39
Saving a model



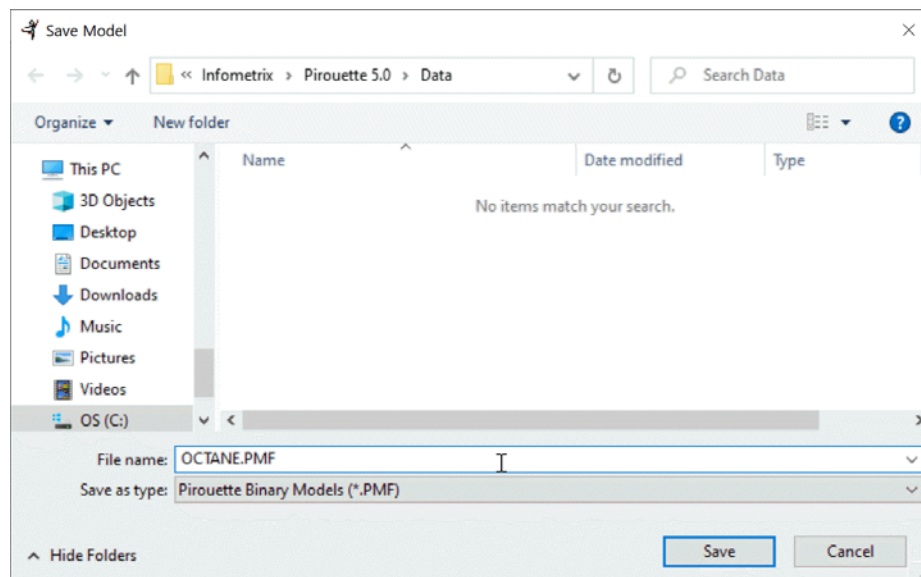
and the model save dialog box will come to the screen. As implied above, the action of saving a model requires three decisions:

- Choosing which model;
- Choosing a name for the file; and
- Choosing a model format (binary or ASCII)

For the purposes of this exercise,

- Select the second PLS model run on the No Outlier subset in the list at the bottom of the dialog box

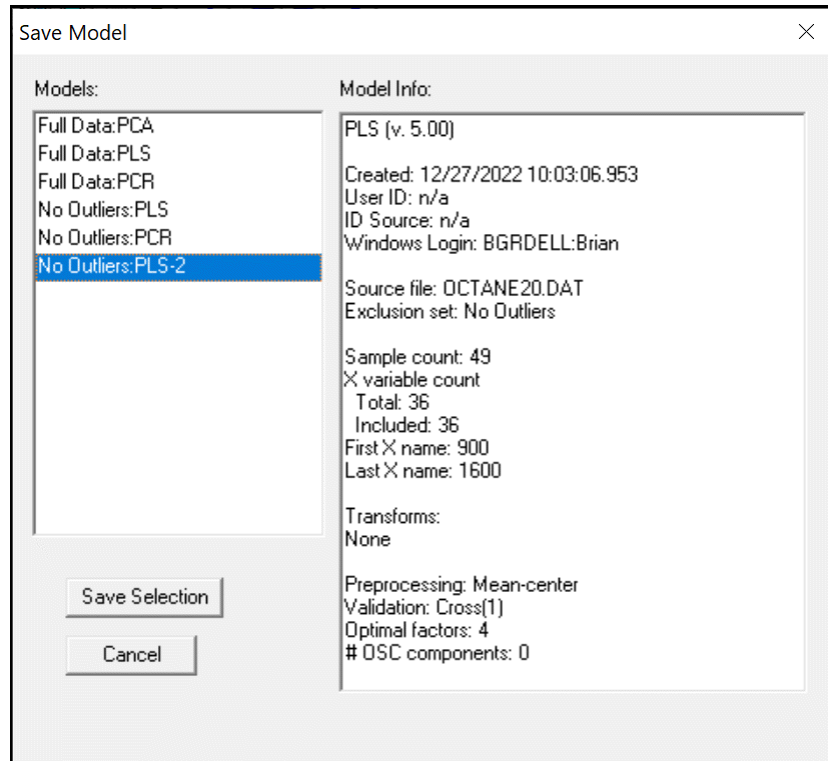
Figure 3.40
Setting model type
and name



- Click on the Second NO Outliers PLS model to select it
- Click on the Save Selection button

Note that clicking on an entry in the Available Models list produces a display of information about that model at the right of the dialog. Choosing save Selection will bring up a standard Windows dialog box.

Figure 3.41
Model information



- Type OCTANE.PMF into the edit field of the dialog box as shown
- Click on the Save button

The PLS model based on 49 samples has now been created on disk and is available for future predictions. The next phase of this walkthrough will use this model file to make predictions on another data file.

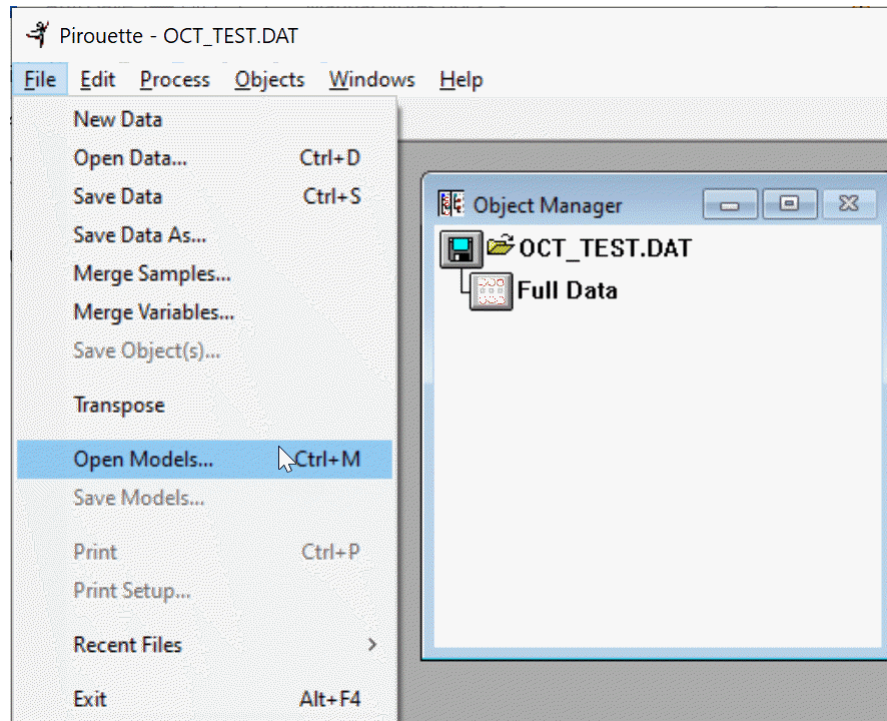
PREDICTION OF UNKNOWN S

A data set has been collected to allow you to test predictions made by the model we have just created. In this section, we will open this new data set, predict the octane values and compare the results to those generated by the octane engine. The test set (OCT_TEST.DAT) contains 10 new spectra of different gasoline samples.

- Choose the Open Data option from the File menu
- Select the OCT_TEST.DAT file from the list of ASCII format files in the DATA sub-directory
- Click OK
- Choose the Open Model option from the File menu

3 Regression Tutorial: Calibration and Model Validation

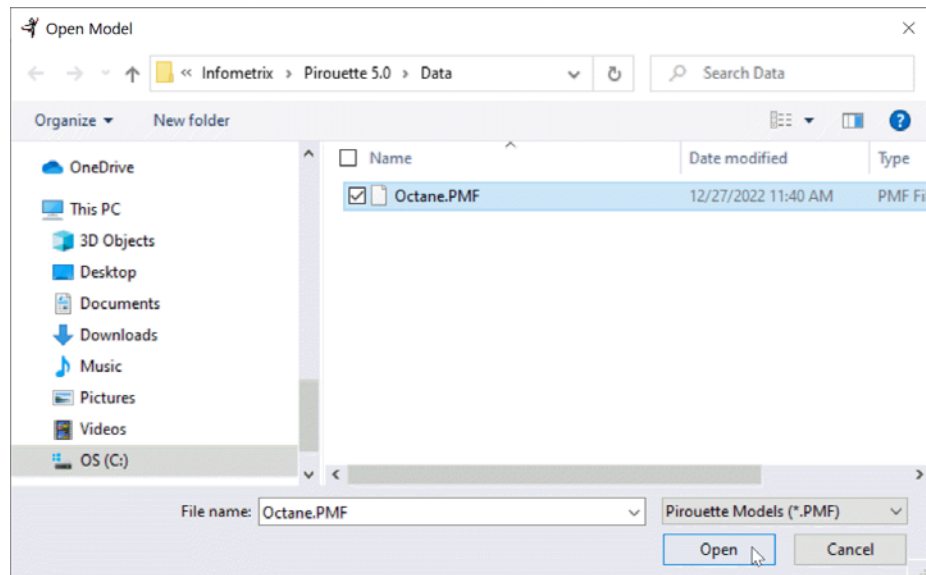
Figure 3.42
Opening a model file



The Open Model dialog box will appear on the screen as shown below, giving you the choice of models created in this and in prior analyses.

- Choose the OCTANE . PMF model file by clicking once on the name with the mouse
- Click on OK to load the model into Pirouette

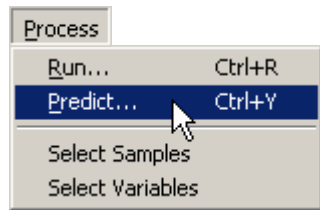
Figure 3.43
The Open Model
Dialog Box



To predict the octane values of these 10 samples,

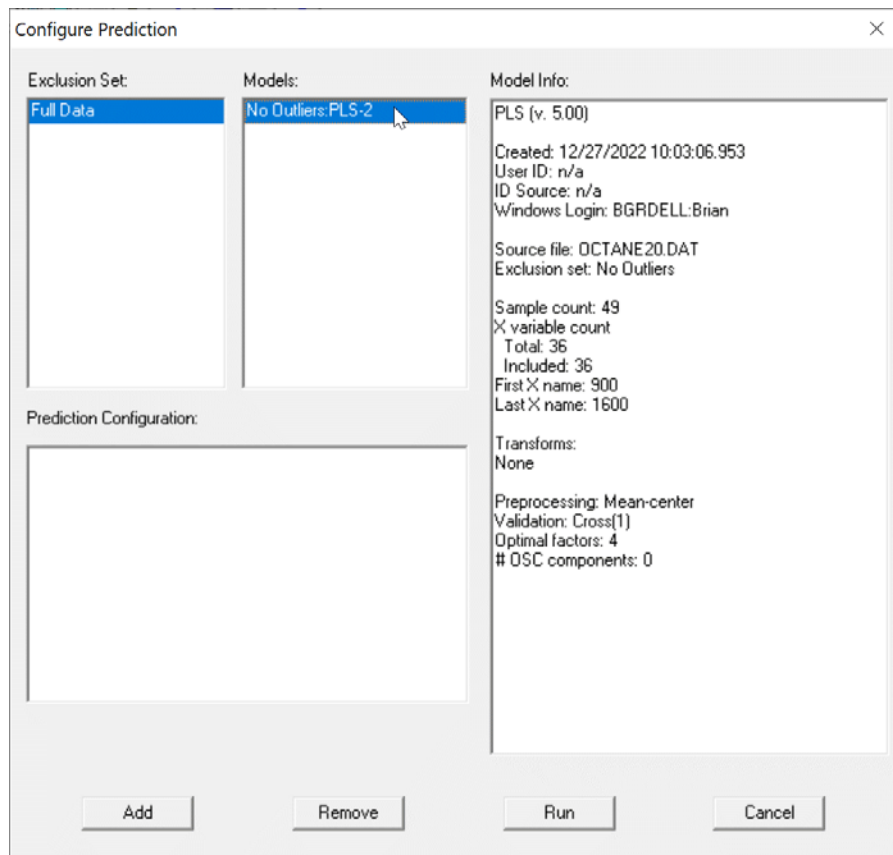
- Select the Predict option from the Process menu

Figure 3.44
Selecting the Predict option



A dialog box will appear on screen that will allow you to configure the prediction much as algorithms are configured for processing within Pirouette. This configuration is shown in the next figure.

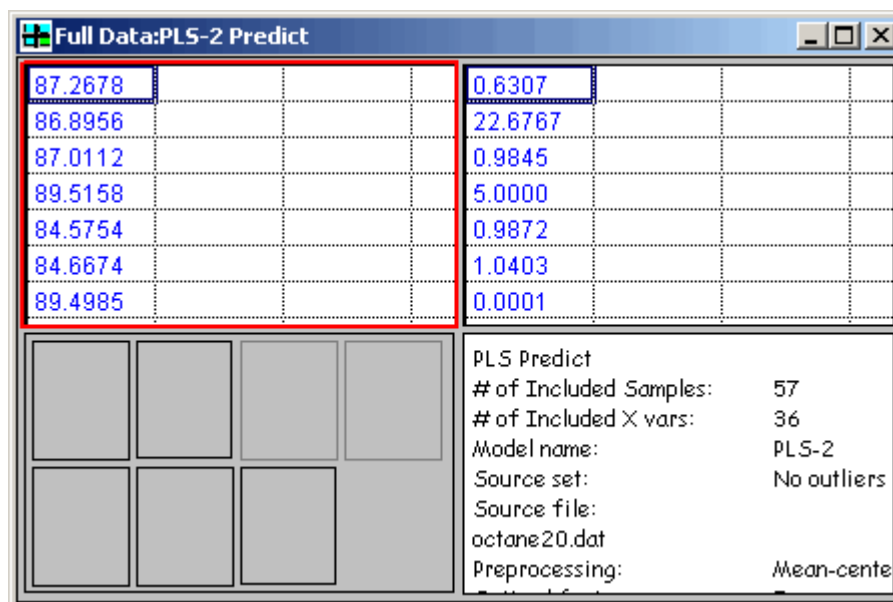
Figure 3.45
Configuring the PLS prediction



- Make the selections of exclusion set (in this case the Full Data) and the model
- Click on Add to create the run list
- Click on the Run button to process

Just as in running the calibration algorithms, the prediction results will appear as a new window of the Pirouette display, listing a single octane value for each sample. Had we saved both a PLS and a PCR model for the original octane data, both predictions would appear. Note that you do not need to specify mean centering as a preprocessing option; the Pirouette model contains this information, so the centering is done automatically.

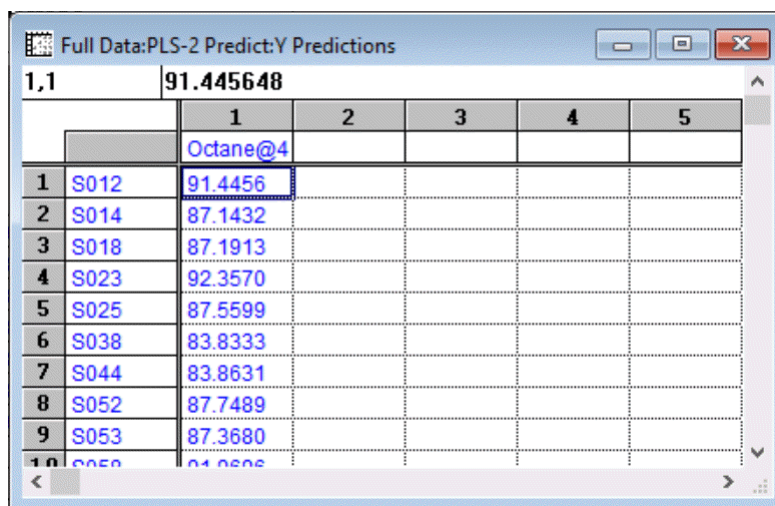
Figure 3.46
Results of the PLS prediction



The results appear in a four-section display. On the top side are the prediction values plus a table giving the parameters of the prediction (SEP, PRESS, r and # of factors). The lower left section contains graphics showing diagnostics such as the Y-Fit, X-residuals and probabilities (see “Running PCR/PLS” on page 7-14 for a more complete explanation of the diagnostics).

- Double-click on the upper left window to zoom the table with the predicted octane results

Figure 3.47
Tabulated PLS prediction results



These results can be compared to the engine values which are contained in the OCT_TEST data file.

- Click-drag the Y Fit object to make a new window
- Click on the Table icon in the ribbon to convert the plot to a tabular view.

The comparison is shown in the following figure.

Figure 3.48
Octane predictions
and residuals using
a PLS model

1,1		91.699997				
		1	2	3	4	5
		Measured	Predicted Y	Residual Y	Upper Lim	Lower Lim
1	S012	91.7000	91.4456	0.2543	91.8335	91.0578
2	S014	87.3000	87.1432	0.1568	87.5348	86.7517
3	S018	86.9000	87.1913	-0.2913	87.5935	86.7892
4	S023	92.2000	92.3570	-0.1570	92.7434	91.9705
5	S025	87.8000	87.5599	0.2401	87.9591	87.1608
6	S038	83.7000	83.8333	-0.1333	84.2173	83.4494
7	S044	83.9000	83.8631	0.0369	84.2462	83.4800
8	S052	87.2000	87.7489	-0.5489	88.1305	87.3673
9	S053	87.4000	87.3680	0.0321	87.7599	86.9760
10	S058	91.9000	91.9696	-0.0696	92.3616	91.5776
11						

The Y-fit object contains a column with measured values if they exist in the file. So, column one above lists the values measured in the lab for these gasoline samples, column 2 shows the result of PLS predictions, and column 3 shows the difference between the two. The last two columns provide an estimate of the error bar for the PLS predictions. With the analysis complete, you can quit Pirouette or proceed to the analysis of new data.

Review

In this section, we used spectra to classify gasoline samples based on the presence or absence of additives, then created and optimized a model of the outlier-free data to be used in subsequent predictions. We have shown that factor-based methods (such as PLS and PCR) can competently model data collected on unleaded gasoline samples for the purpose of predicting the pump octane rating of gasoline.

REFERENCES

1. *Annual Book of ASTM Standards*, (1985) Vol.05.04.
2. Kelly, J.J.; Barlow, C.H.; Jinguji, T.M. and Callis, J.B., *Anal. Chem.* (1989) 61(4): 313-320.
3. Rohrback, B.G., *Trends in Anal. Chem.* (1991) 10(9): 269-271.

Part II.

Guide to Multivariate

Analysis

- 4 Preparing for Analysis**
- 5 Exploratory Analysis**
- 6 Classification Methods**
- 7 Regression Methods**
- 8 Mixture Analysis**
- 9 Examples**

Preparing for Analysis

Contents

Overview	4-1
Defining the Problem	4-3
Organizing the Data	4-4
Checking Data Validity	4-5
Visualizing the Data	4-6
Transforms	4-10
Preprocessing	4-26
Calibration Transfer	4-33
Final Remarks	4-35
References	4-36

Multivariate data analysis, as powerful as it may be, can be daunting to the uninitiated. It has its own jargon and incorporates concepts which at first glance seem strange. Typically, a novice user becomes comfortable with the multivariate approach only after a period of confusion and frustration. Why bother? The rationale behind multivariate data analysis is simple: univariate methods, while well-understood and proven for many applications, can sometimes produce misleading results and overlook meaningful information in complex data sets.

Univariate methods were developed for univariate data. Today, however, data are routinely acquired from spectrometers and chromatographs, multichannel instruments which generate many measurements for each sample analyzed. Applying univariate methods to such data is tantamount to discarding all but one of the measurements. While some problems may yield to a thorough statistical analysis of a single variable, this approach has several drawbacks when applied to multivariate data. First, it is tedious to look for the one needle in a veritable haystack of variables. Second, it is incomplete if multivariable relationships are important. Thus, a multivariate approach is mandated by the structure of the data set. If the measurements are numerous and/or correlated, processing them as a unit is advised.

Overview

Proper data preparation is integral to a successful multivariate analysis. The wise user will address a number of issues before embarking on the computational portion of the

project. This chapter suggests a strategy for planning a multivariate analysis and then details ways of preparing data. Of the steps involved in performing a multivariate analysis listed below, the first five are addressed in this chapter.

- **Define the problem** - Surprisingly enough, problem definition is an often neglected part of the analysis process. The purpose of collecting an infrared spectrum is not to create and store a piece of paper. Instead, it may be to determine if the sample is of acceptable quality. Before looking at the data, decide what questions you hope to answer. Examine the scientific literature or consult local experts who can offer suggestions about your proposed activities. Perhaps your problem has been posed before and you can learn from previous investigations.
- **Organize the data** - Devise a strategy for assembling and organizing the necessary data. Your project requires a collection of samples. How many are available? Will more become available later? If you need to combine various data sources, are the file formats compatible?
- **Check data validity** - Verify that the computerized version of the data agrees with the original measurements. Decide how to handle missing data values.
- **Visualize the data** - Examine the data with graphical presentations. For example, use scatter plots to look for trends, clusters, correlation and invariance. Extremely unusual samples are apparent in such plots. You may also get a feel for the structure of the data.
- **Transform/preprocess the data** - It is often necessary to “adjust” a data set before running a multivariate algorithm. For example, variability due to sample size differences in chromatography can be minimized by normalizing each sample. Transforms such as differentiation can emphasize spectral features and compensate for drift. Scale and range differences are addressed by preprocessing. When one variable’s magnitude is much larger than others, this variable alone may dominate subsequent computations based on variance or distance. Sometimes this dominance is appropriate and no preprocessing is necessary. Often, however, some form of preprocessing is warranted to mitigate the influence of variable ranges and magnitudes.
- **Perform exploratory analysis** - Exploratory analysis is the computation and graphical display of patterns of association in multivariate data sets. Exploratory algorithms reduce large and complex data sets to a suite of best views. They give you insight into sample or variable correlations. Pirouette’s two exploratory algorithms, Hierarchical Cluster Analysis (HCA) and Principal Component Analysis (PCA), are discussed in [Chapter 5, Exploratory Analysis](#). Do an exploratory analysis on every problem because the process is relatively quick. Besides exposing possible outliers, it can tell you if your data possess sufficient modeling power to warrant further investigation.
- **Create a model** - A fundamental reason for collecting data is to develop predictive classification and regression models which characterize future samples. Classification in Pirouette is the computation and the graphical display of class (or category) assignments based on multivariate similarity. Pirouette’s three classification algorithms, K-Nearest Neighbors (KNN), Soft Independent Modeling of Class Analogy (SIMCA), and PLS-Discriminant Analysis (PLS-DA) are discussed in [Chapter 6, Classification Methods](#). A large number of applications involve predicting a difficult to measure value from easier measurements by using a regression algorithm to establish the relationship between the difficult and easier measurements. Pirouette’s two factor-based techniques, Principal Component Regression (PCR) and Partial Least Squares (PLS), are discussed in [Chapter 7, Regression Methods](#), as is the Classical Least Squares (CLS) algorithm. PLS and PCR can also be used within Pirouette for locally weighted regression (LWR). Finally, there are two regression-based unmixing algorithms: Alternating Least Squares (ALS) and Multivariate Curve Resolution (MCR) discussed in [Chapter 8, Mixture Analysis](#).

- **Examine computed results** - After running an algorithm, examine carefully its results. They tell you if the analysis was successful or pinpoint reasons for its failure. Compare these results with your expectations and develop new perspectives on the underlying meaning in your data.
- **Validate the model** - To determine the reliability of a model, it is necessary to include a validation step. Although a model can be used for prediction immediately after its creation, whenever possible we recommend that its reliability first be investigated by a process called validation. Validating PCA models is treated in [“Model Validation” in Chapter 5](#); see [“Validation-Based Criteria” in Chapter 7](#) for a discussion on validating regression models.
- **Use the model** - Compare the results with previous expectations. Are they in agreement? Has your knowledge improved as a result of the analyses? Interpretation cannot be supplied by the computer. A human analyst is ultimately responsible for ascribing meaning to results.
- **Update the model** - If the model is in use for some time, or is to be used on another instrument or in another laboratory, is it still performing reliably? If necessary and feasible, it may be necessary to totally replace the data used to create the model. If this is not possible, the model may still be useful by applying a calibration transfer (see [“Calibration Transfer” on page 4-33](#)).

Defining the Problem

Begin by deciding the general nature of your project. Is it an exploratory problem, a classification problem, a regression problem or a combination of these types?

In some data analysis scenarios, no prior knowledge regarding trends, groups or relationships exists, suggesting the need for exploratory analysis. Here the basic consideration is whether an inherent structure in the data implies relationships amongst samples and/or variables. Predefined class assignments for samples or measurements of properties of interest may not be available. Part of exploratory analysis is a search for sample groupings which might stimulate a need for more data collection. Another part is the identification of unusual samples. The importance of the variables in groupings and variance patterns can also be investigated. Example exploratory questions are:

- Can my chromatography results distinguish region of origin for olive oils?
- Which process line measurements are affected most by batch-to-batch variation?

In some instances, the issue is to differentiate among discrete categories, a classification or pattern recognition problem. Here problem definition becomes more distinct. The key concerns are how accurately can we distinguish between categories and which variables are most important for the distinctions. Example classification questions are:

- Can we routinely identify bacteria genus and species from fatty acid profiles?
- Is it possible to use trace organic analysis to match and determine fire accelerant source in arson cases?

Finally, the goal may be to determine composition or indirectly assess a physical/chemical property, which is a regression problem. Here the focus is on accuracy, precision, and, as always, the importance of variables. Example regression questions are:

- Is it possible to use a spectrometer to predict water content in cheese to within 0.5%?
- Can we monitor extent of polymerization as the product is forming?

Once you have determined your problem type, you are ready to inquire into the data history: how and when were they generated, what types of measurement methods were used, what levels of precision pertain to each variable, etc. Have previous analyses been performed? Does pertinent *a priori* information exist? How were the samples stored? Are the most recent measurements different from those taken earlier? Has the instrumental performance changed during the sample collection process? Was experimental design implemented? Obviously data quality impacts the ultimate quality of conclusions drawn from data analysis. It is always a good idea to know as much as possible about the procedures which gave rise to the data. Having said this, we must confess that sometimes the questions posed above are addressed only after multivariate analysis suggests that something more is “going on” than was first thought.

Organizing the Data

It is usually most convenient to place all data into a single file which may require concatenating data from a variety of sources. The advantage of this accumulation is clear: the more samples and variables, the better your chance of understanding the chemical/physical processes underlying the data. Disadvantages exist as well: data from disparate sources can be difficult to assemble and unwieldy to manipulate. Be advised that getting all of your data into a single computer-readable file is often the most time-consuming step of the project.

ASSEMBLING THE PIECES

Pirouette allows you to assemble large data sets through a merge facility—adding either new samples or new measurements on old samples. Moreover, you can paste data into a Pirouette spreadsheet via the Clipboard; see [“Cut, Copy, Paste, and Clear” on page 13-9](#) for details. Also, investigate the export formats supported by your instrument software. Devising a smooth path from data generation to data congregation takes time, but once you learn some “tricks”, you can speed up this process significantly.

Some questions to consider during the initial organizational effort include:

- Are the data from an instrument? If so, are they from more than one instrument or type? What is the precision of the instrument(s)?
- Are there questionnaire or hand-recorded data?
- Are data from each sample in single files?
- Are class and/or dependent variable(s) data stored in separate files or will they be hand-entered?

Data can be hand-entered into the Pirouette spreadsheet; see [“Changing Data Values” on page 13-8](#). They can be imported from Excel files or from existing ASCII files; see [“Opening and Merging Existing Data Files” on page 14-3](#). Other formats are also allowed. If class or dependent variable values are stored separately, use the File Merge function discussed in [“Opening and Merging Existing Data Files” on page 14-3](#), after insuring that the samples in both data sets are the same and in the same order. Once all data have been assembled, any class or dependent variables must be designated as such; see [“Changing Variable Types” on page 13-10](#). Why this is necessary is explained in the next section.

When combining data from one or more sources for multiple samples, keep in mind that Pirouette defines a sample (or object or case) as a line (or row) of information. A row can

contain a sample name, class variables (one or more category designations) and measured variables. Measured variables can be multichannel instrument responses (*e.g.*, absorbances), separation results (*e.g.*, peak heights or areas at specific or relative retention times), multiple specific assays from single channel instruments or physical/chemical/biological tests. Non-instrumental data may be included (*e.g.*, data from a sensory or expert panel). Measured variables can be either dependent (y) or independent (x).

TRAINING SET STRUCTURE

The data structure required by modeling algorithms (most commonly PCA, KNN, SIMCA, CLS, PCR and PLS) merits some explanation. Building either a classification or regression model requires a training set, also called an experience set. A training set contains more than the independent variables on which a model is based.

If a classification model is to be built, the extra information is a class variable, *i.e.*, the *a priori* category assignment for each sample. For example, if the goal is to categorize samples as either fresh or stale based on NIR spectra, then each sample in the training set must have already been determined to be fresh or stale by some means unrelated to NIR spectrometry, and this information must be included along with the spectra. Thus, running KNN, SIMCA, or PLS-DA requires a data set containing at least one class variable.

If a regression model is to be built, the extra information is a dependent variable, *i.e.*, the parameter to be predicted for each sample. For example, if the goal is to predict the percentage of protein in wheat from NIR spectra, then each sample in the training set must have already been analyzed for protein, perhaps by a wet chemical method, and this information must be included along with the spectra. Thus, running either PCR or PLS requires a data set containing at least one dependent variable.

PLS-DA is a special case in which the algorithm uses the information in a class variable to form multiple dependent variables.

Checking Data Validity

After the data have been assembled in one file, viewing it in the Pirouette spreadsheet is next. Quickly scroll through the samples and variables to find missing values (signified by asterisks), especially if you have merged data from several sources. If data sets with different dimensionality are merged, Pirouette fills empty cells in incomplete rows or columns with asterisks to maintain a rectangular data area.

To deal with missing values you can:

- Supply values, either row-wise or column-wise
- Exclude rows or columns containing missing values
- Delete rows or columns containing missing values

If only a few values are missing, you may apply one of Pirouette's fill options; see [“Filling Missing Values” on page 13-13](#). Filling is recommended only when relatively few values are missing.

If many values are missing, it is more appropriate to exclude the affected samples and/or variables; see [“Creating Subsets from Tables” on page 13-20](#). In this way, they have no effect, except through the influence of removing an important variable. Later, you may be able to collect the data that were lacking and repeat the Pirouette analyses with the variable included.

Sometimes you may decide that the missing values should be permanently removed from the data set, particularly if you anticipate never recovering the sample or repeating a measurement. In this situation, a logical choice is to delete sample(s) and /or variable(s) with missing entries before proceeding; see “[Insert and Delete](#)” on page 13-9.

If you decide to exclude a variable because of missing values, you can proceed with most analyses without problems. However, you will not be able to apply some transforms. Some Pirouette transforms (discussed in “[Transforms](#)” on page 4-10) involve all variables. For example, taking derivatives and performing smooths are prohibited when missing values are present. In these cases you must either delete variables with missing values or fill them.

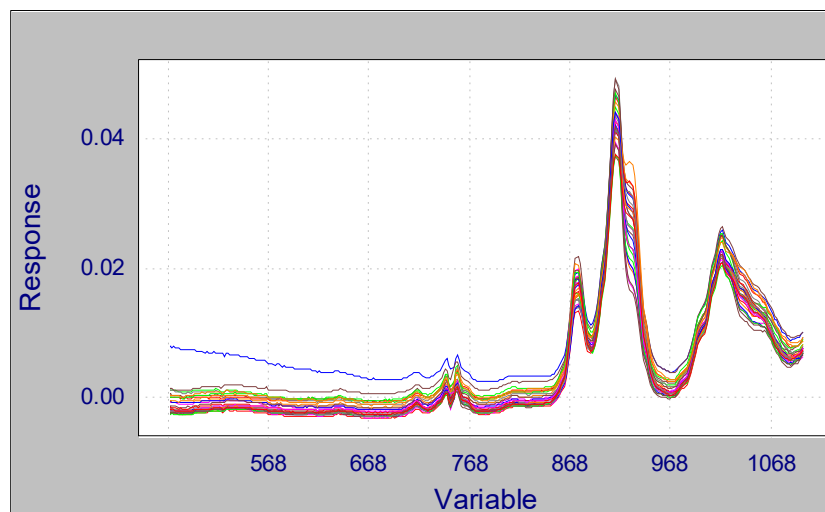
Visualizing the Data

Plots of the raw data convey its structure. The plot views shown below can point out important data features. They provide a taste of the graphical approach embraced by Pirouette. Remember, a picture is worth a thousand words/numbers. A table of numbers is more easily comprehended graphically. As you become a more experienced Pirouette user, you will discover your own ways to visualize data and occasionally find a view offering a new and instructive perspective. See “[Pirouette Graph Types](#)” on page 12-4, to become familiar with Pirouette’s plot types.

LINE PLOTS

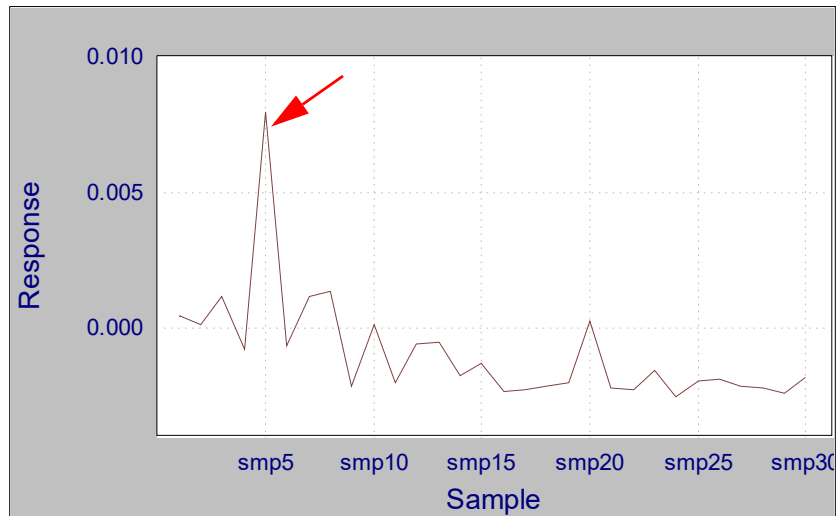
A line plot of a sample may indicate anomalous values that you might overlook in scanning the numbers in the spreadsheet. Overlaying line plots of all samples can give a quick indication of specious samples—samples which may be outliers. The bulk of the data in the following figure are fairly homogeneous but one sample is obviously distinct.

Figure 4.1
Line plot of data with
one unusual sample



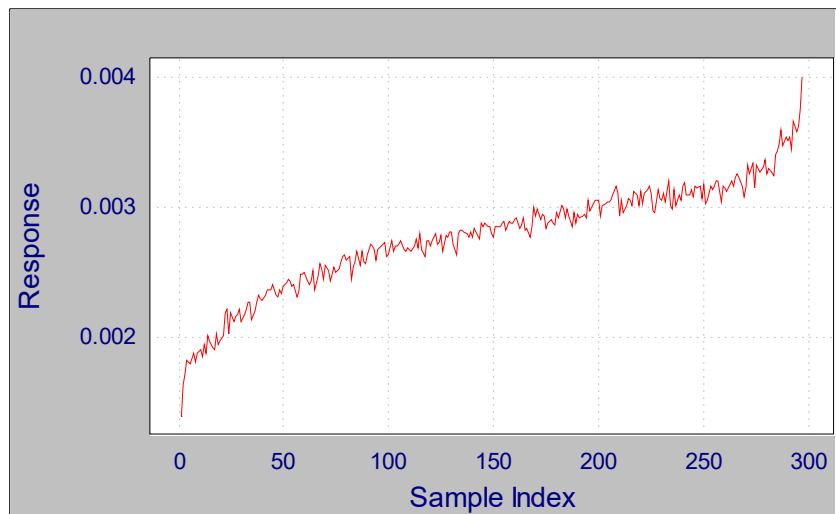
Line plots of variables can also reveal obvious anomalies. In the next figure, the outlier sample seen in [Figure 4.1](#), which is the fifth sample, stands out when the data are presented in a line plot of a variable.

Figure 4.2
Line plot of a variable
with one unusual
sample



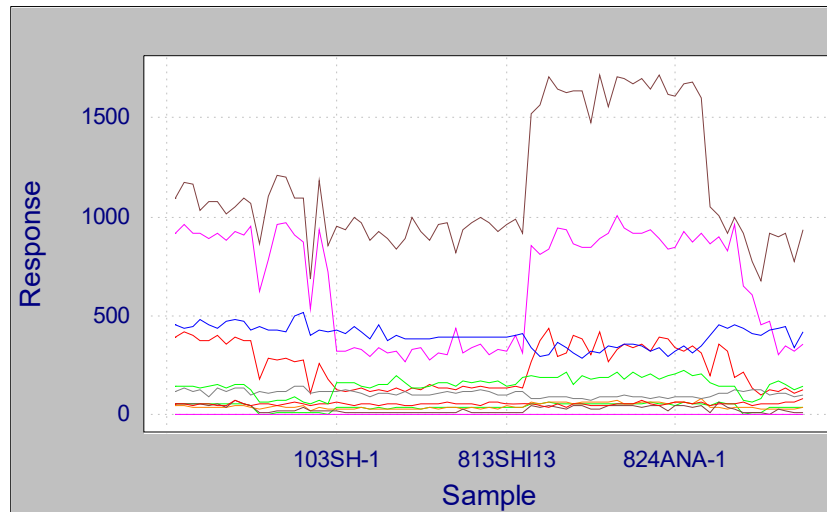
A line plot of a single variable can also reveal trends in the data. For example, a steadily decreasing value for a variable across the sequential order of samples may be expected or it may indicate an undesirable drift in the measurement. Such a decrease occurs in the figure below. You must decide if this behavior is acceptable.

Figure 4.3
Line plot of a variable
with a trend



Similarly, overlaid line plots of variables might reveal other trends or anomalies. For this view to be of value, the plotted variables must span the same relative range of response. Otherwise, some plotted information is buried close to the baseline and difficult to assess. The next figure is an example of an overlaid line plot of several variables, showing the different response levels of the variables, as well as different relative responses among subgroups of samples in the set.

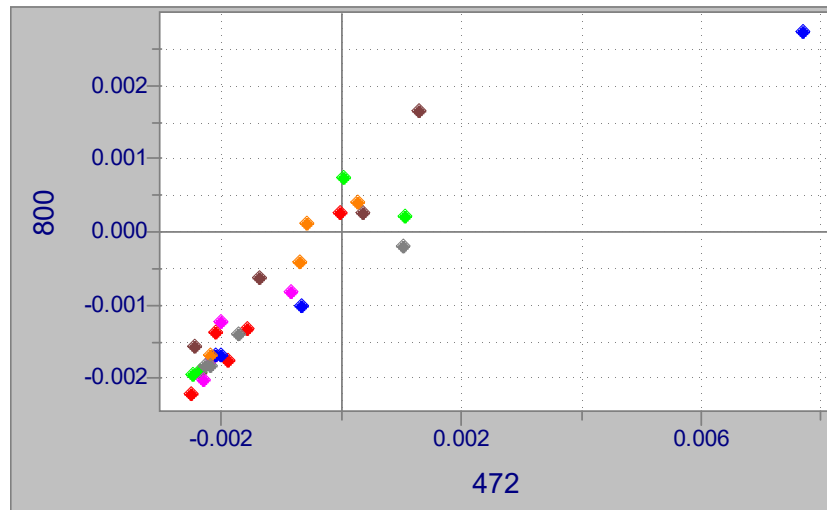
Figure 4.4
Line plot of several variables



SCATTER PLOTS

Variable scatter plots, either 2D or 3D, may reveal correlations between variables. Unusual samples stand out in a scatter plot. For example, in the following figure, an outlier is clearly visible.

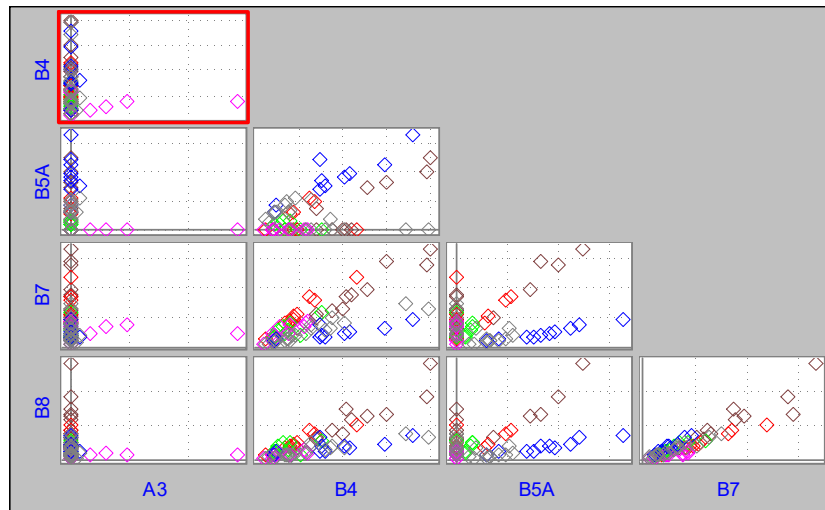
Figure 4.5
Scatter plot with obvious outlier



A 3D scatter plot has the added advantage of showing three variables simultaneously. Couple that with the ability to rotate the points in 3 dimensions and this presentation becomes very forceful.

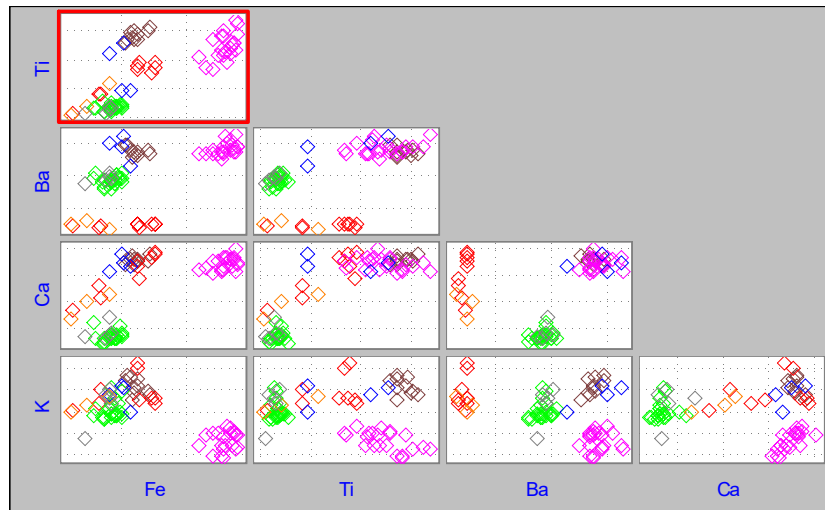
A multiplot is a collection of 2D scatter plots. Pirouette's multiplots facilitate the often tedious task of generating and inspecting many 2D bivariable combinations. The following multiplot indicates that some variables are highly correlated while others are nearly invariant.

Figure 4.6
Multiplot with
correlated and
uncorrected
variables



With luck, you may uncover a set of variables which clearly distinguish sample clusters. If your goal is classification, this visual analysis may suffice. The multiplot below contains subplots in which clusters are obvious (*e.g.*, K vs Ba).

Figure 4.7
Multiplot with
distinguishable
clusters in several
subplots



Of course, examining multiplots is feasible only when the number of variables is relatively small, say, less than 20. Otherwise, the number of subplots would make it impractical to show in a single multiplot because there would be too little plot area in each individual subplot.

Although there are many ways to visualize the structure in a data set, all are ultimately limited by our ability to process only two or three dimensions of information at one time. Because multivariate data sets have by definition high dimensionality, computational approaches have been developed to overcome this limit on our ability to interpret and characterize complex data sets. Often, the result of such computations is to isolate the relevant information in only a few derived variables. These techniques are the subject of the remaining chapters in this section of the manual.

Transforms

For some data types, it helps to transform the independent variables (the x block) prior to analysis. This is especially true when the independent variables consist of like measurements which vary in time (as in a chromatogram or a cyclic voltammogram) or wavelength (as in a spectrum). For example, there may be noise spikes in a signal stream. Smoothing can lessen the effects of these random variations. Alternatively, a spectrum may contain subtle peak shoulders enhanced by computing a derivative.

VIEWING TRANSFORMED DATA

You should **always** have a definite reason to apply a transform and **always** view its effects on the data. To accomplish this:

- Go to Process/Run
- Select XFORM in the list of Algorithms
- Move the desired transforms from the Available list into the Selected box, in the desired sequence
- Click on an exclusion set to which the transforms will be applied
- Click on Add
- Continue to configure transform sequences and exclusion sets
- Click on Run

Running the XFORM algorithm tells Pirouette to apply the selected transforms to the configured exclusion set; drag this computed object to the work area to display a window containing a plot of the transformed data.

When you are convinced that the transform settings have a reasonable and useful effect and wish to apply them prior to the execution of an exploratory, classification or regression algorithm, confirm that they are in the Selected box when you add the algorithm. When algorithms are run, any selected transforms are applied before preprocessing; see [“Preprocessing” on page 4-26](#) for a discussion of this topic. As row-oriented operations, transforms are applied to each included row (thus, samples) separately. Consult the individual discussions below for the details concerning specific transforms.

CONFIGURING TRANSFORMS

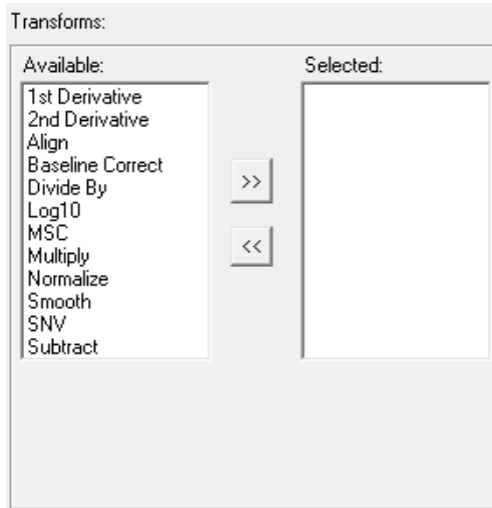
In Pirouette, the available transforms are:

- 1st and 2nd Derivative
- Align
- Baseline Correction
- Divide by functions
- Log 10
- Multiply
- Normalize
- Multiplicative Scatter Correction
- Smooth

- Standard Normal Variate
- Subtract

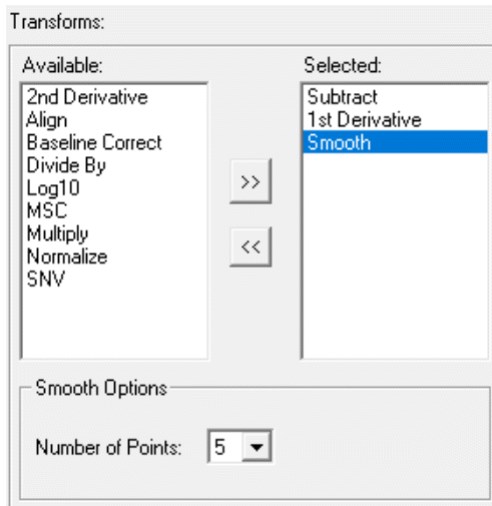
Transforms are specified by moving them from the Available list box to the Selected list box in the Run Configure dialog; see the figure below. Several offer customizable parameters which are displayed when the transform name is clicked with the mouse.

Figure 4.8
Transforms portion
of Run Configure
dialog box



Note that you can apply a sequence of transforms. For example, in the figure shown below, three are specified. First, the value of the variable #1 is subtracted from each sample, then the first derivative is taken, then the result is smoothed. No transform, however, can be applied more than once.

Figure 4.9
Applying three
transforms
sequentially



Using a Mask

Several transforms can employ a mask to further customize their action. A mask is a row in the data set containing only zeroes and ones. Variables containing values set to one are used by the transform, and those set to zero are not. How those variables are to be used depends on the transform.

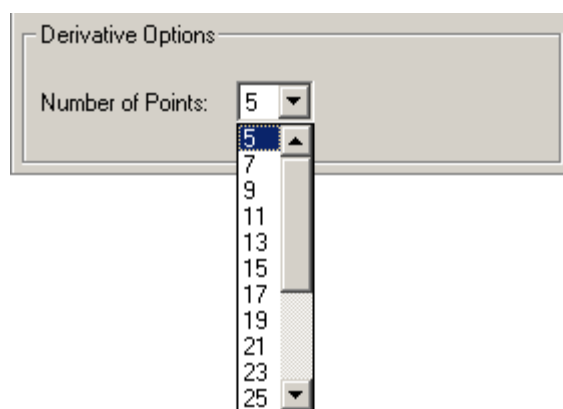
4 Preparing for Analysis: Transforms

To create a mask, insert a new row (see “Insert and Delete” on page 13-9) in the spreadsheet of the data. An inserted row is automatically excluded; since it is not real data, just an aid for Pirouette for computing transforms, it should stay excluded. Initially, you may want to place ones in the appropriate cell ranges (see “Filling Missing Values” on page 13-13), then fill the row with zeroes.

Derivatives and Smoother

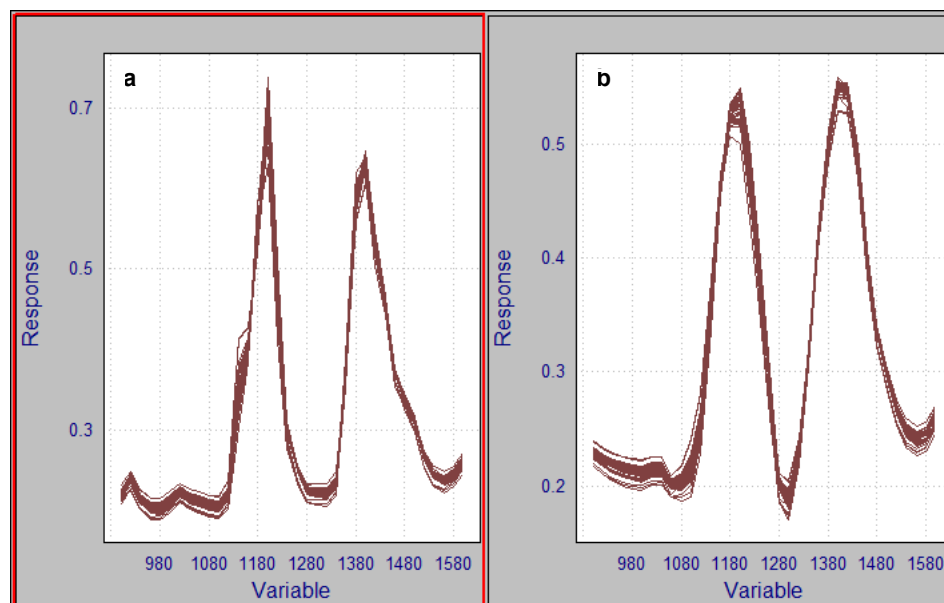
The 1st and 2nd derivative and smoothing transforms are based on a Savitzky-Golay polynomial filter¹. This method applies a convolution to independent variables in a window containing a center data point and n points on either side. A weighted second-order polynomial is fit to these $2n + 1$ points and the center point is replaced by the fitted value. The three transforms differ in the weighting coefficients. The filters implemented in Pirouette include the modification suggested by Gorry² to handle the first and last n points.

Figure 4.10
Specifying number
of window points



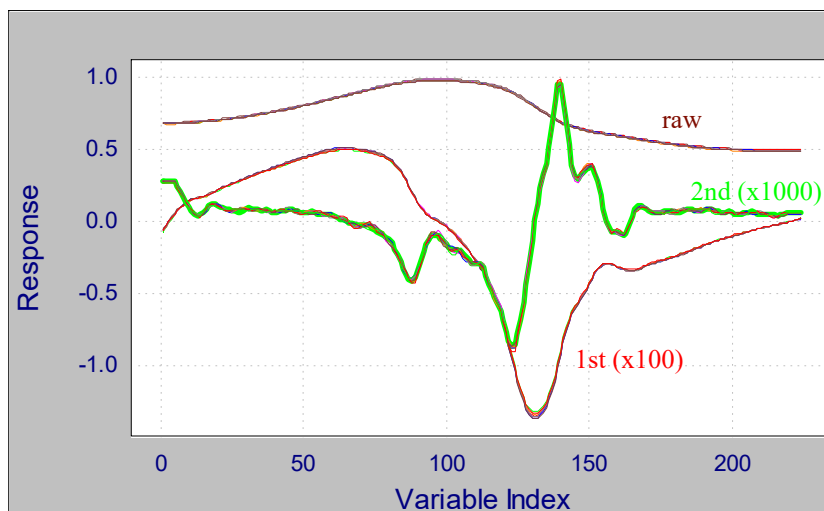
You choose the number of window points in a smooth or derivative via the associated drop down list box (see above). The number of window points must be less than the number of independent variables; otherwise the run aborts. The next figure shows the effect of a nine point smooth.

Figure 4.11
(a) Before smooth (b)
After smooth



The effects of the nine point derivative are shown below. Note that each successive derivative reduces the signal magnitude, as well as the signal to noise, a consideration if your raw data signals are not strong.

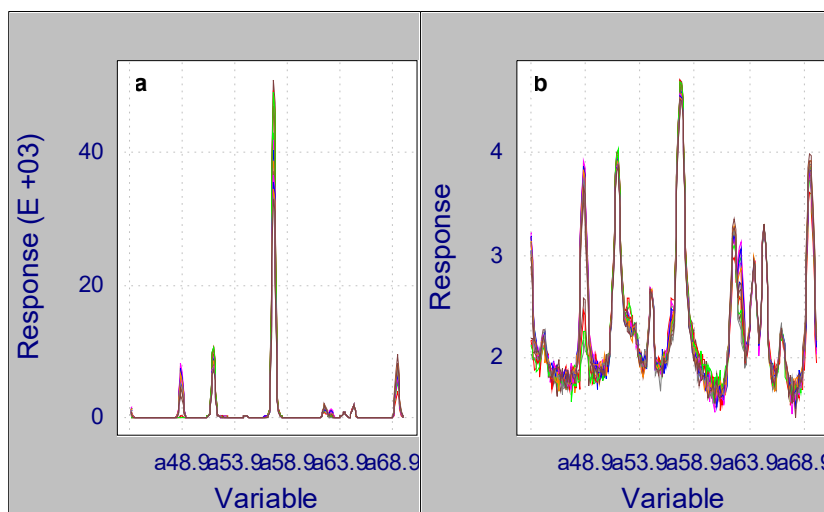
Figure 4.12
Raw data contrasted
with 1st and 2nd
derivatives.



Log 10

This transform computes the base 10 logarithm of the absolute value of each independent variable. It emphasizes small data values relative to larger ones. An example is shown below. When the log of zero is taken, the result is -6.92369, the log of the smallest positive number which can be represented in single precision.

Figure 4.13
(a) Before log
(b) After log



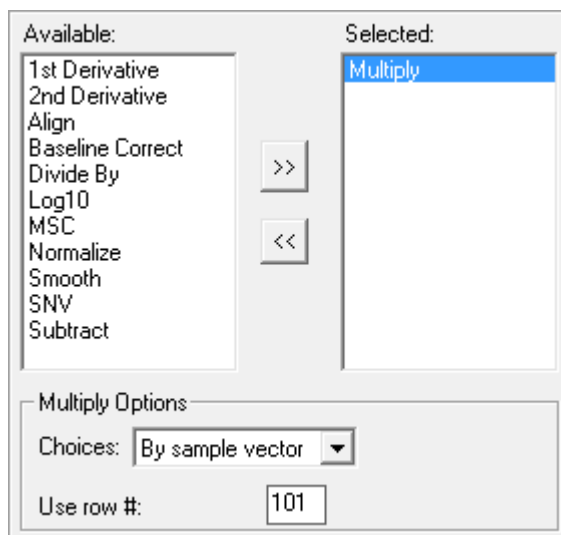
Note: Because transmittance (optical spectroscopy) does not vary linearly with concentration, transmittance values should be converted to absorbance before regression models are developed to predict concentration. Although Pirouette does not provide a direct Transmittance-Absorbance conversion, the functional equivalent is a Log 10 transform followed by Multiply by -1.

Multiply

The basic Multiply transform is straightforward: supply a constant by which to multiply all values. See the note above for a common usage in spectroscopy. Multiply can also be useful when all data values are extremely large or small, for example, when a second derivative decreases the magnitude of the data significantly or for data from some NMR instruments whose values are in the millions.

It is also possible to multiply every value in a row by a corresponding value in a specified vector. This might be useful when you want to correct the magnitudes of measures that are considerably different among the variables.

Figure 4.14
Multiply by Vector

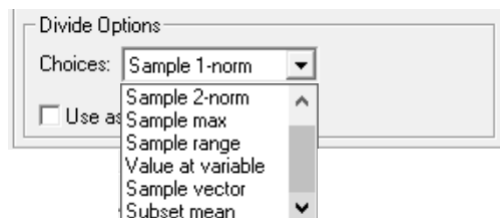


This complements the Divide by Vector transform, discussed below (“Divide by Sample Vector” on page 4-17).

Normalize/Divide By

Many definitions are given for normalization. All share the idea of dividing each data value by a normalization factor. The various Divide By options (shown below) provide flexibility in computing the normalization factor for the i th sample, f_i . Each included variable in that sample is then divided by f_i .

Figure 4.15
Divide By options



Divide By Sample 1-norm

This is also known as *area normalization* because the normalization factor is simply the area under the sample profile, the sum of the absolute values of all included variables for sample i :

$$f_i = \sum_j^{m^*} |x_{ij}| \quad [4.1]$$

The symbol m^* indicates included variables.

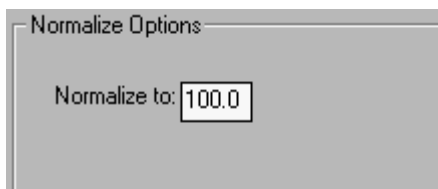
Divide By Sample 2-norm

This is also known as *vector length normalization* since each included variable is divided by the length of the sample vector:

$$f_i = \left(\sum_j^{m^*} x_{ij}^2 \right)^{1/2} \quad [4.2]$$

Note that this transform is also available as the Normalize option:

Figure 4.16
Normalize scaling
parameter



which includes a scaling parameter applied after the division.

Note: *Divide By, 2-norm is equivalent to Normalize to 1. Normalize remains in the list of transforms for backward compatibility.*

Divide By Sample Max

Dividing by the sample's *maximum* value scales data in a manner typical for mass spectrometry, in which the most abundant mass fragment is given a value of 100%. Thus, the normalization factor is simply:

$$f_i = \max(x_i) \quad [4.3]$$

Divide By Sample Range

When data derive from different instruments, such as chromatography systems from different vendors, not only can the baselines differ, but the actual data units might not be the same. An easy way to put these measurements on a comparable scale is to divide by the *range* in values across the sample. This normalization factor is then:

$$f_i = \max(x_i) - \min(x_i) \quad [4.4]$$

This transform differs slightly from the others in that the sample's minimum is subtracted from each value before dividing by the normalization factor:

$$x(norm)_{ij} = \frac{x_{ij} - \min(x_i)}{f_i} \quad [4.5]$$

In general, the Divide By Sample options just described differ from other transforms in that they depend on which variables are used in computing the normalization factor, *i.e.*, normalization with all variables included has a different effect than normalization with several variables excluded. This dependence is most notable if the excluded variables are relatively large in magnitude. Normalizing is most appropriate when response depends on sample size—for example, in chromatography with mass-sensitive detection.

Using a Mask

Sometimes it is desirable to compute the normalization factor using one set of variables and have it applied to a different set. The mask row permits this scenario. Create a mask (see “Using a Mask” on page 4-11) and set to one those variables which are to determine the normalization factor. The variables with values = 1 can be excluded or included. Then, in the Transforms dialog box, indicate the mask row number. The appropriate normalization factor is then computed for each sample from variables having ones in the mask. All included variables in that sample are then divided by the normalization factor.

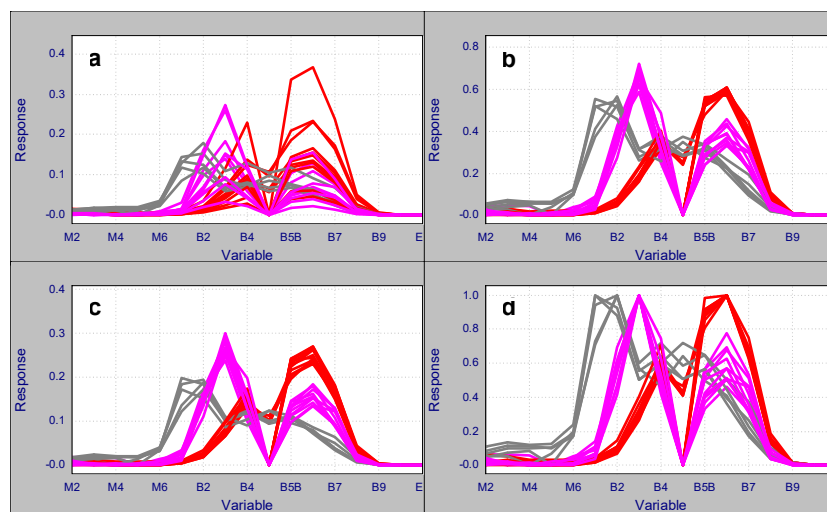
Divide by Value at Variable

In absorption spectrometry, it may be desirable to normalize to the *value at a particular variable n*, so that the normalization factor is:

$$f_i = x_{in} \quad [4.6]$$

Some examples of the results of the Divide By transforms are given below.

Figure 4.17
(a) Raw data;
after divide by the
(b) 2-norm;
(c) 1-norm;
(d) max norm



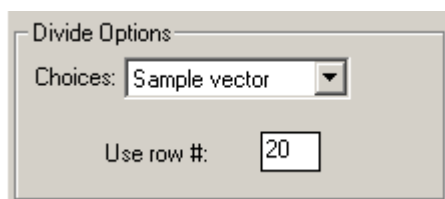
Note: *Because a normalized vector has a length of 1, we have effectively “removed” one independent value from the sample. If we know $m - 1$ (*i.e.*, all but one) of the values, we can compute the remaining value from the vector length. This phenomenon is known as closure. Closed data can behave erratically under certain conditions, especially if there are very large and very small values and if there are relatively few variables. Thus, normalizing data containing fewer than ~ 10 variables is not advised³.*

Divide by Sample Vector

When data are merged, as variables, from sources in which the measurement units are considerably different--such as from spectroscopy and chromatography--the variables from the source with the smaller unit may have values that will swamp those from the other source. Scaling by variance would not properly correct this disparity. It is possible, however, to insert a scale factor with this Divide by transform, a form of block scaling.

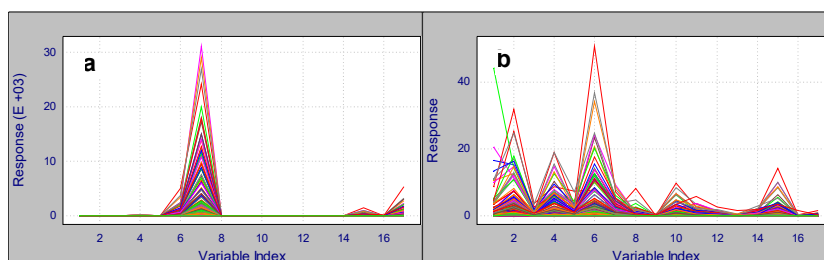
Create an excluded row to be used as a vector of divisors and insert values into the appropriate variables to control the normalization factor for the variables. Then, in the Transforms dialog box, indicate the row number.

Figure 4.18
Divide by Sample
Vector option



The appropriate normalization factor for each variable is drawn from the corresponding value in the indicated row. An example of this situation is shown below.

Figure 4.19
(a) Raw data, and (b)
after Divide by
sample vector

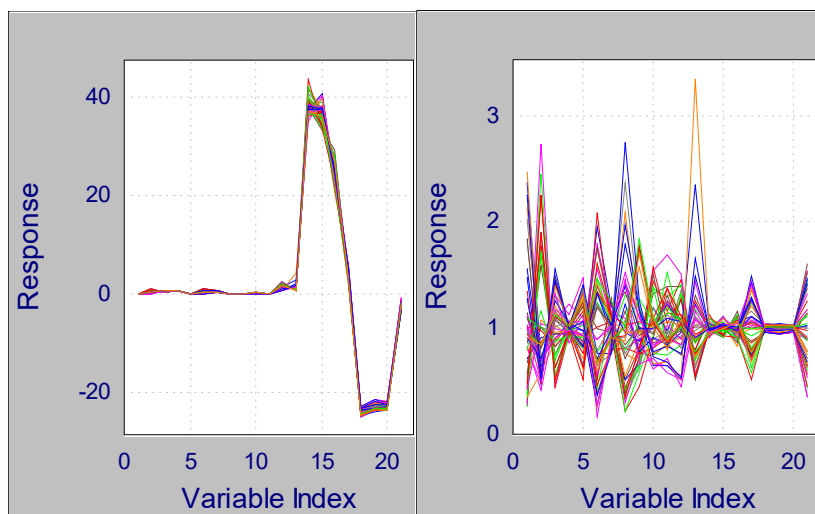
**Divide by Subset mean**

For some data sources, the variables represent measures with different units. Biomarker data from geochemical measures provide an example. This can have the inadvertent effect of enhancing the importance of those variables whose units make numbers seem large when, in fact, all variables should be given the same importance. If possible, it is a good idea to scale the all variables to have approximately the same numeric range. Autoscaling is one way to accomplish this but has the side effect of emphasizing noisy variables and produces positive and negative values.

An alternative to autoscaling is to divide each variable by a constant such that all variables will be approximately on the same scale. This can be done before entering the data into Pirouette but can be done as well by creating a sort of dummy variable that contains the scale factors, then telling Pirouette to “Divide by Sample Vector”.

Another more automated way to do this is to use the Divide by Subset mean option. In this case, the mean of each variable will be used as a divisor for that variable, where the mean will be taken for all rows that are included in the subset. An example of using the Subset mean for scaling variables is shown below. Note the additional benefit to this approach: the transformed data show all positive values.

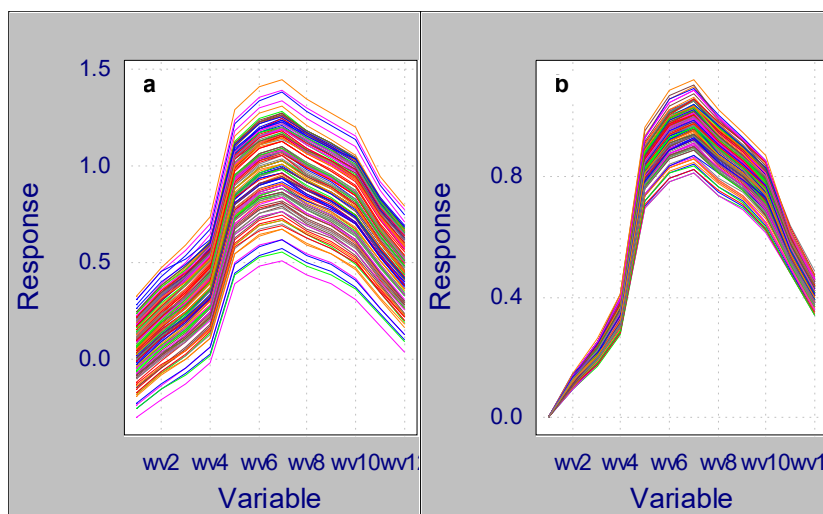
Figure 4.20 Raw data and after Divide by Subset mean



Subtract

A limited form of background removal can be accomplished with the Subtract transform. In Pirouette, Subtract functions in two ways: by subtracting a user-specified constant from all independent variables or by subtracting the value at one variable from the remaining variables for each sample. In the first case, you can choose a positive or negative value. An illustration of the effect of variable subtraction is given below.

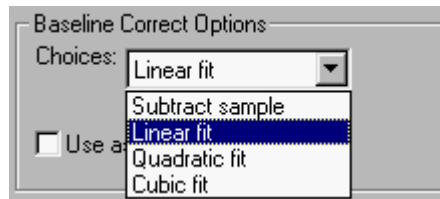
Figure 4.21
Subtract variable #1
(a) before (b) after



Baseline Correction

The Baseline Correction transform corrects offsets by subtracting a profile, rather than a single point. This profile can be a row in the data table or can be derived from a curve fit. These choices are shown below.

Figure 4.22
Baseline correction
options



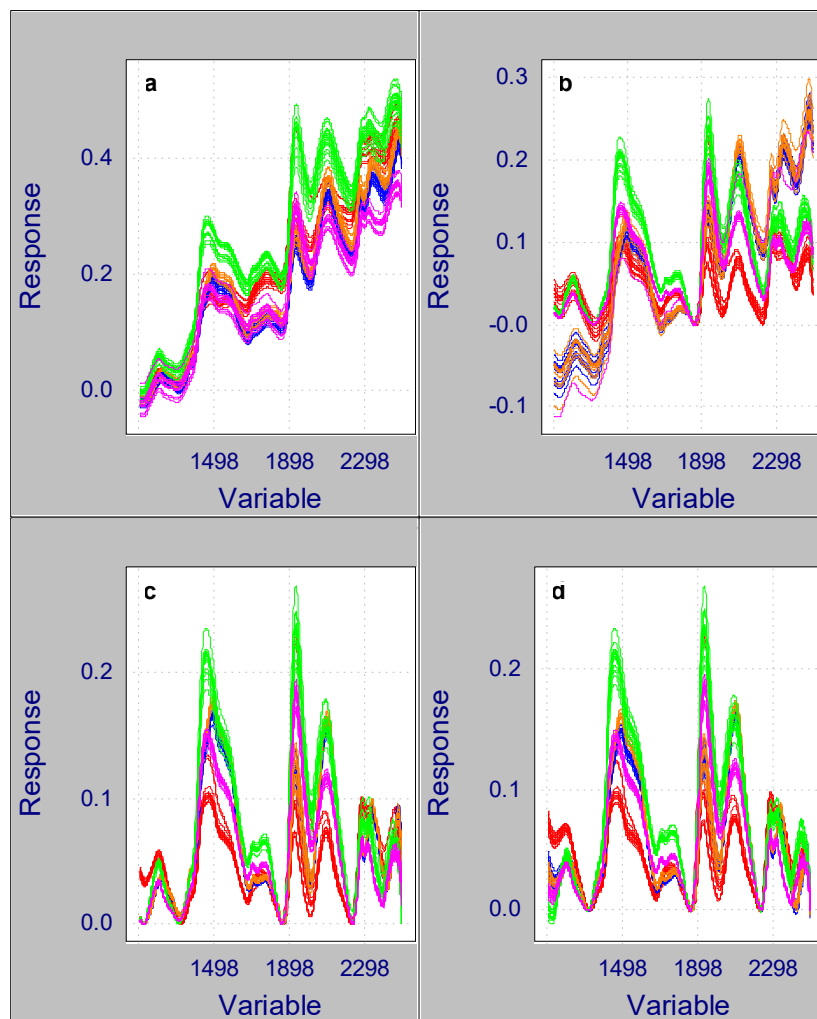
Curve Fitting a Profile

Finding a baseline for each sample requires knowledge about which variables make up the baseline and the degree of polynomial to be fit. Linear, quadratic and cubic fits employ a first, second and third degree polynomial, respectively. The most straightforward way to specify baseline variables is via a mask containing ones for every variable considered “in the baseline”. See [“Using a Mask” on page 4-11](#).

Avoiding the mask creation process and supplying only the degree of fit invokes an iterative method of finding baseline variables. First, all included variables are used to compute a trial baseline of the specified degree. Only variables with points lying on or below the trial baseline are retained as baseline variables and another trial is computed. As this process repeats, fewer variables remain in the baseline. When only 1% of all included variables are left in the baseline or when the number of baseline variables does not decrease, the process stops and the actual baseline is computed on the retained variables. Thus, which and how many variables are determined to be “in the baseline” varies from sample to sample.

4 Preparing for Analysis: Transforms

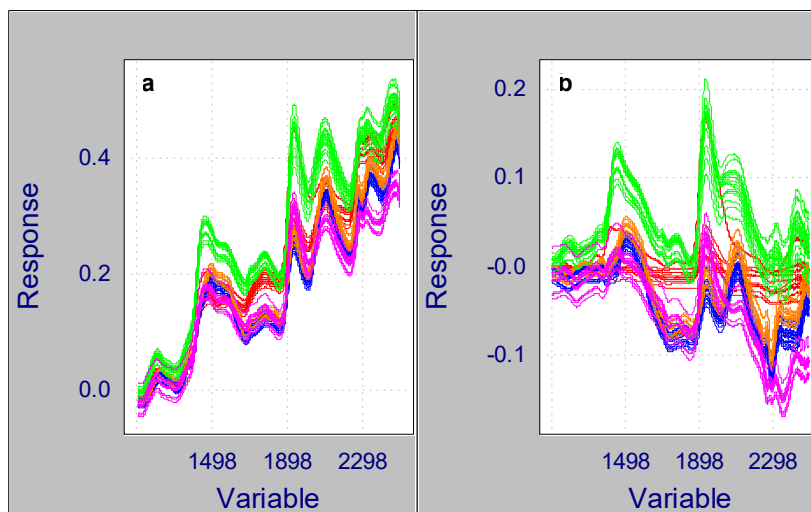
Figure 4.23
Baseline correction:
(a) Raw data;
(b) Linear fit;
(c) Quadratic fit;
(d) Cubic fit



Subtract sample

To subtract the same row in the data table from every sample, supply the row number as shown in the figure below. The indicated row must be excluded from the subset being transformed, otherwise the run will abort.

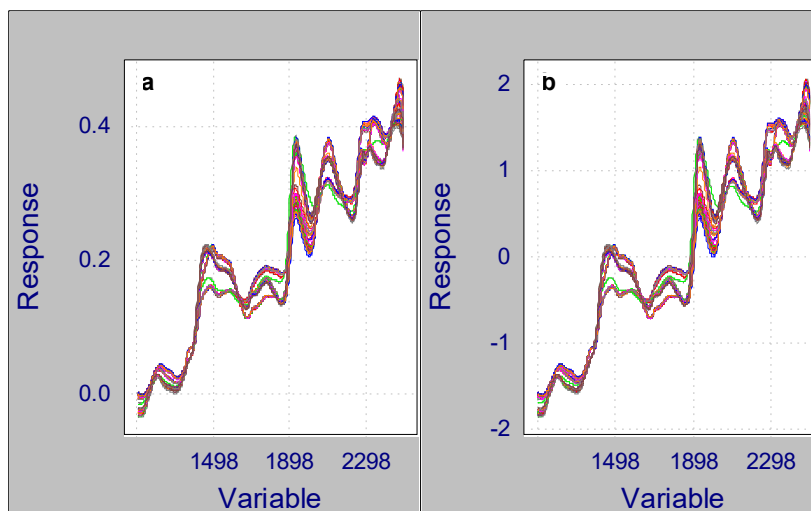
Figure 4.24
Baseline correction
(a) Raw data, (b)
Subtract selected
sample



MSC

MSC (Multiplicative Scatter Correction) is a standard approach to compensating for scattering by solids in NIR spectrometry. Each sample spectrum is regressed linearly against an ideal spectrum to yield a slope and intercept. The sample spectrum is then “corrected” at each wavelength by first subtracting the intercept, then dividing by the slope. The ideal spectrum is most often simply the mean of included samples. Defining the mean spectrum as the ideal is sometimes cited as a disadvantage of MSC. However, there is no general agreement on what might more properly constitute the ideal spectrum. Be aware that MSC violates the general rule in Pirouette that transforms affect samples independently. Because the mean spectrum is calculated from the included samples, the correction of a sample varies with the number and spectra of included samples.

Figure 4.25
Examples of (a) MSC,
and (b) SNV
transforms



When MSC is run on a subset with excluded variables, the regression step is performed piece wise, that is, for each region of included variables. Also, when a mask is specified

in MSC, only those variables with ones in the mask are used in the regression step. Some chemometricians have suggested that only wavelengths with no chemical information be used to find correction factors in MSC. This can be achieved applying the two steps just described.

SNV

SNV (Standard Normal Variate) is another approach to compensating for scattering by solids in NIR spectrometry. It can be described as *row-autoscaling*. The mean and standard deviation of a sample are first computed based on included variables; the value for each included variable is corrected by first subtracting the mean, then dividing by the standard deviation. The result is often similar to MSC; see [Figure 4.25](#). One advantage of SNV over MSC is that SNV does not depend on the specific data set being processed; each sample is corrected independently (MSC does its correction relative to the subset mean).

Align

Raw chromatographic data often suffer from a lack of retention time reproducibility across samples, which can severely hinder multivariate analysis. Chromatographic software may compensate for these shifts with internal standards. The retention time of analytical peaks is compared to that of an internal standard (or marker peak), and a correction factor derived from a calibration run adjusts the analytical peak's elution time. Typically this approach is applied to peak data only. The *Align* transform allows a similar correction for the entire profile.

The likelihood of a high quality alignment increases when:

- The elution time difference between an analytical peak and marker peak is small
- The marker compound and analytes interact with the stationary and mobile phase in a similar fashion

To satisfy these criteria, long separations mandate multiple markers. Adjusting an analytical peak's profile with a single marker is risky. It requires either an interpolation between the marker time and zero (or a surrogate zero based on the separation system's 'hold up' time) for peaks eluting before the marker or an extrapolation for peaks eluting after.

The familiar Kovats retention index⁴ employs a homologous series of markers—for example, the n-alkanes—to adjust analytical peak elution times from bracketing markers. Linear interpolation is adequate for (linear) gradient liquid chromatography and most electrophoretic methods. A retention index system can be based on any set of marker compounds. In the analysis of petroleum hydrocarbons, for example, the linear polynuclear aromatic hydrocarbons—benzene, naphthalene, anthracene, tetracene, pentacene—are reasonable markers. In the analysis of vegetable oils, methyl esters of the straight chain fatty acids of various lengths are often used as markers⁵.

Pirouette's alignment implementation requires that suitable marker retention times are stored in the Y-block; they form the basis for the adjustments of the whole chromatograms comprising the X block. The markers must, of course, be present in all samples, including any calibration or reference samples. Their values are assumed to have units of scan variable number, not actual retention time.

Several file formats read by Pirouette understand the idea of a Y variable. Input of data in these formats avoids having to enter manually the marker peak values. For example,

one flavor of the Pirouette AIA file read extracts named peak information and assigns their scan times to the Y block (see [page 14-10](#)).

Note: To take advantage of this special AIA file read, the user should configure the chromatographic method to report only named peaks and to create a calibration file containing only peaks that will be used as alignment markers.

Finding Markers Manually

If the marker positions are not available in the file containing the profiles, then the user must enter the marker values manually. First, create as many Y variables as markers. Next, decide which sample in the data set is the calibration sample, that is, to which sample will all other samples be aligned. Then, visually scan a line plot of this calibration sample to find the variable # closest to the first marker's retention time, then type this value into the first Y column. Continue to find marker variable #'s for the remaining Y variables for the calibration sample.

Figure 4.26
Example data with
anchor values set for
a calibration sample

1256	1257	1258	1259	1260	Y1	Y2
627.5	628	628.5	629	629.5	a1	a2
0.0034	0.0034	0.0035	0.0035	0.0035	380.0000	1105.0000
0.0027	0.0027	0.0027	0.0027	0.0027	0.0000	0.0000
0.0033	0.0033	0.0034	0.0034	0.0034	0.0000	0.0000
0.0021	0.0022	0.0022	0.0022	0.0022	0.0000	0.0000
0.0030	0.0030	0.0031	0.0031	0.0031	0.0000	0.0000
0.0032	0.0032	0.0032	0.0032	0.0032	0.0000	0.0000
0.0026	0.0026	0.0027	0.0027	0.0027	0.0000	0.0000
0.0027	0.0028	0.0028	0.0028	0.0028	0.0000	0.0000
0.0034	0.0035	0.0035	0.0035	0.0035	0.0000	0.0000
0.0036	0.0036	0.0037	0.0037	0.0037	0.0000	0.0000
0.0039	0.0039	0.0039	0.0039	0.0039	0.0000	0.0000

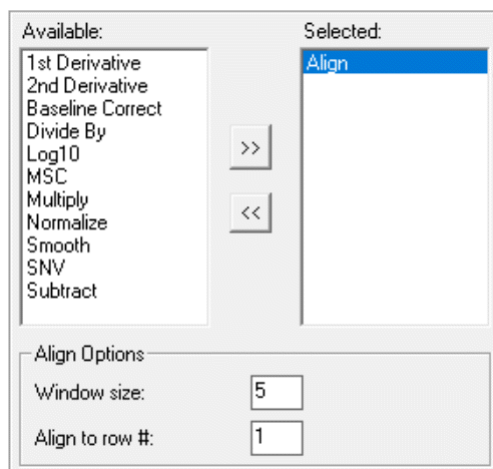
You could continue this process for all samples to be aligned. But, the process can also be automated by entering zeroes for non-calibration samples. When a non-calibration sample marker is zero, the calibration sample marker value is used as a starting point for finding the precise anchor value for the sample.

If the window size option (see below) is too small, the true peak top will not be found, and the alignment results will be in error, manifested by marker peaks in the samples which are not in alignment with the calibration sample. Adjust the window size or the estimated marker peak top position and rerun the Align transform until all sample marker peaks align.

Align Options

Only two options need be set to run the Align transform, and these are shown in the accompanying figure.

Figure 4.27
Align options



The Align To Row # specifies the row in the dataset to which all other profiles are aligned, that is, it points at the calibration sample. This row can be included in or excluded from the subset being aligned.

The Window Size is key to automating exact marker location. Recall that the user initially specifies a nominal elution value (in terms of X block index) for each marker by inserting the index value in the Y block of the spreadsheet. A more precise value is found by first searching around the nominal value for a local maximum, then refining that local maximum by fitting a parabola to 5 points centered on the point. The Window Size determines the region that will be searched for the local maximum; choose a value on the order of an average peak width.

As an example, consider a chromatogram with 500 time units and peaks with widths of about 12 time units. A nominal marker value of 427 and a Window Size of 11 means that the local maximum would first be found by examining the signal at variables 422-432. If the signal maximum occurs at variable 423, the refined maximum would then be determined by fitting a parabola through the signals at 421-425.

A Window Size of 0 is a special case and indicates that no refinement of the supplied values is to be done. Thus, the marker values inserted in the spreadsheet will be used as is for the adjustment calculations. This is useful when it is necessary to iterate with the Align parameters because you can apply data in the Anchor object from a previous run, copying them into the Y-block marker cells.

An example data set, named MYCALIGN.DAT, is supplied so that you can follow these steps on a small file. In this data set, two anchors (the chromatographic internal standards) are present in all of the samples to be aligned as well as in the calibration file (labeled as a QC sample). The anchor times are already included in the file; examine a line plot of the data to locate these values relative to other peaks, before and after alignment.

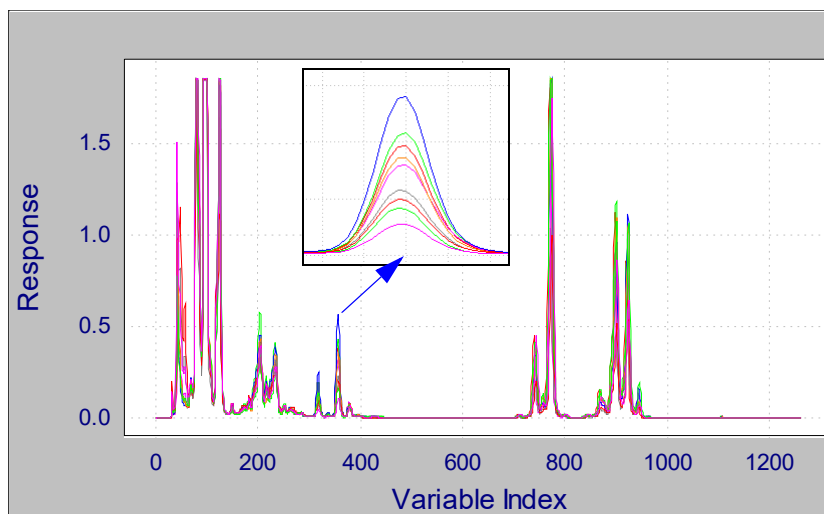
Align differs slightly from the other transforms, in that it produces two objects: the aligned X block profiles and the updated marker times/positions, which are called Anchors. The Anchors contain the fitted peak maxima for each marker in each sample. If you initially set the sample markers to zero, to use the calibration markers as surrogates, you may want to save the computed anchor times for future reference.

Figure 4.28
An Anchors object

Full Data:LINEUP:ANCH				
28,7		1	2	3
		a1	a2	
1	QCPOS023	376.9127	1105.5876	
2	A19977CA	377.6656	1103.8794	
3	A14472CA	377.7373	1105.7494	
4	V01159CA	377.7135	1106.1218	
5	V01358CA	378.1770	1106.4752	
6	I51063CA	379.0171	1107.6962	
7	I51078CA	378.5953	1107.3422	
8	I51113CA	378.2153	1107.1836	
9	I51220CA	379.2963	1108.7307	
10	I51235CA	378.7484	1108.3735	
11	I51260CA	379.3690	1108.7485	

The aligned profiles compose a matrix that is the exact size of the data set (or exclusion set) matrix. The line plot display gives a good visual indication of the alignment achieved by showing the anchor points as exact overlays. An example of a set of aligned profiles is shown below, together with a blown-up region around one of the anchors.

Figure 4.29
Aligned profiles;
(inset) marker peak



Saving Aligned Results

Because the alignment process happens as part of a transform, subsequent analysis, such as PCA or HCA, can be performed on automatically aligned data. Nevertheless, you may still want to preserve the aligned data in a separate file for later use. To export aligned profiles, use the Save Objects command under the File menu. If the data were originally exported as AIA files from a chromatography system, you may want to choose the AIA Raw Data file type filter (see [page 15-5](#)).

Note: Infometrix offers a companion software package, called *LineUp*, which does not require markers. To learn more about this product visit [LineUp](#) on the web.

Preprocessing

Preprocessing, in contrast to transforms, is a column-oriented operation so preprocessing results are specific to a set of samples. Adding a sample to a data set can greatly influence the effect of a preprocessing technique. The difference between preprocessing data on a variable-basis and transforming data on a sample-basis is important to appreciate. Preprocessing is necessary because several multivariate algorithms compute results driven by variance patterns in the independent variables. Sometimes arbitrary measurement scales and/or magnitudes produce misleading results when an inappropriate preprocessing is made. A series of illustrations will make this clear.

MEAN-CENTER

Relationships among samples are more easily visualized by placing the plot origin at the center of the data set. Thus, points are often centered about the mean. For example, consider the following data.

Table 4.1
Example raw data of
palm fruits

	Iodine Index	Oleic Acid
Tucumã	76.0	65.67
Dendé	58.0	37.00
Inajá	74.9	48.72
Burití	68.5	72.81
Patauá	75.1	80.20
Pupunha	56.6	53.56

A mean is computed for each variable via the following formula:

$$\bar{x}_j = \frac{1}{n} \sum_i^n x_{ij} \quad [4.7]$$

The mean is then subtracted from each data value to produce a mean-centered matrix:

$$x_{ij(mc)} = x_{ij} - \bar{x}_j \quad [4.8]$$

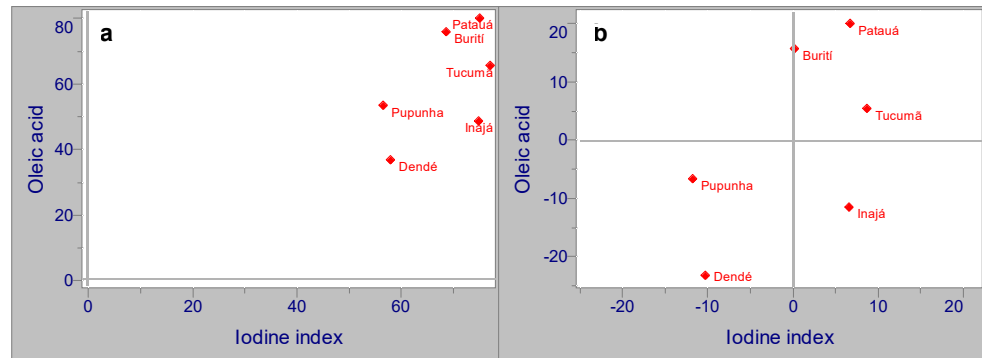
Thus, we can compute the mean-corrected values derived from [Table 4.1](#); these are shown in the next table.

Table 4.2
Mean-Centered Data

	Iodine Index	Oleic Acid
Tucumã	7.8167	6.0100
Dendé	-10.1833	-22.6600
Inajá	6.7167	-10.9400
Burití	0.3167	13.1500
Patauá	6.9167	20.5400
Pupunha	-11.5833	-6.1000

To illustrate the effect of mean-centering, the data from both tables are plotted as below.

Figure 4.30
Plots of (a) raw data
(b) mean-centered data



Mean-centering is recommended for most data types as it merely shifts the origin without altering relative inter-sample relationships. However, when performing multivariate regression (see [Chapter 7, Regression Methods](#)) on data which vary linearly with concentration, have no baseline, and are not closed, it may be best to avoid the centering step⁶.

VARIANCE SCALE

When data from two (or more) disparate variables span different magnitude ranges, the largest variable dominates any variance computations. Consider, for example, measurements of different properties (*e.g.*, temperature, pH, humidity, *etc*) where the units create arbitrary variable ranges. Thus, a change in pH of one unit (say, from pH 7 to pH 8) could be masked by a change in temperature of 10 units (say, from 300 K to 310 K). In such situations, the dominant variable's masking influence can be removed by variance scaling.

On the other hand, spectral measurements usually exhibit significant correlation among the variables. In this case, there is no masking since the variation patterns for both large and small variables are quite similar. These data are not usually variance scaled.

The table below contains measurements of two variables. In this case, the units are the same (% composition) but differences in magnitude and range cause one variable, oleic acid, to dominate the total variance of the data set.

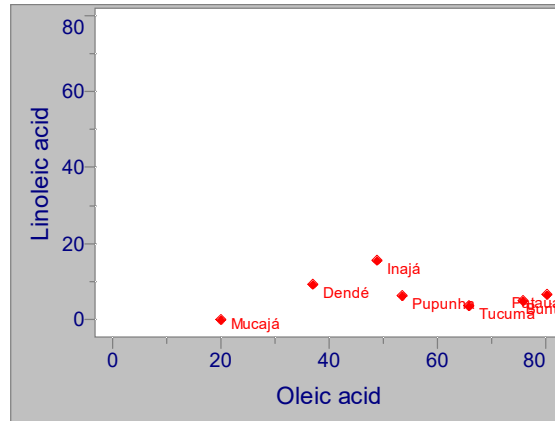
Table 4.3
Variables with
Different Magnitudes

	% Oleic Acid	% Linoleic Acid
Tucumã	65.67	3.65
Dendé	37.00	9.26
Inajá	48.72	15.50
Buriti	75.81	4.86
Patauá	80.20	6.70
Pupunha	53.56	6.27
Mucajá	20.00	0.00

The 2D scatter plot in the next figure shows clearly the dominant variable. If both varied equally, points would be spread over the plot window. That all of the data points are compressed against one axis implies that linoleic acid varies over a much smaller range than oleic acid. If no scaling is applied, variance-based algorithms would “see” Oleic acid as the dominant variable.

4 Preparing for Analysis: Preprocessing

Figure 4.31
Comparing
variances of oleic
and linoleic acid



One way to address this problem is by dividing all values for a variable by its standard deviation. (This approach is called variance scaling although it would be more logical to call it standard deviation scaling.) Thus, we first compute the variance for each variable:

$$s_j^2 = \frac{1}{n-1} \sum_i^n (x_{ij} - \bar{x}_j)^2 \quad [4.9]$$

Then, each independent variable is divided by the appropriate standard deviation, s_j :

$$x_{ij(vs)} = \frac{x_{ij}}{s_j} \quad [4.10]$$

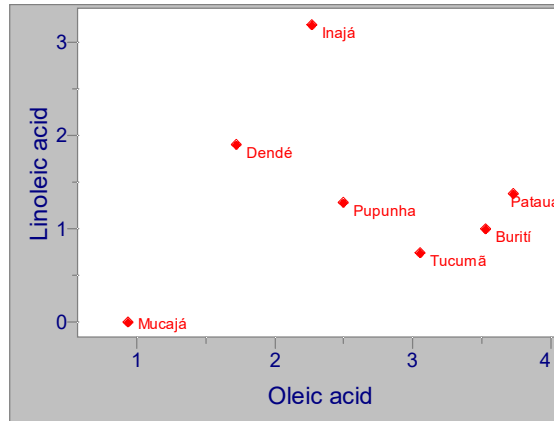
The following table contains the result of variance scaling the data in [Table 4.3](#).

Table 4.4
Variance Scaled Data

	Oleic Acid	Linoleic Acid
Tucumã	3.05628	0.75109
Dendé	1.72198	1.90551
Inajá	2.26743	3.18956
Burití	3.52819	1.00008
Patauá	3.73250	1.37871
Pupunha	2.49268	1.29023
Mucajá	0.93080	0.00000

These data are plotted in the next figure; the points are more equally distributed along both axes. Contrast this plot with [Figure 4.31](#).

Figure 4.32
Variance scaled data



AUTOSCALE

Autoscaling finds use in many fields including statistics. For these applications, autoscaling is simply mean-centering followed by variance scaling:

$$x_{ij(as)} = \frac{x_{ij} - \bar{x}_j}{s_j} \quad [4.11]$$

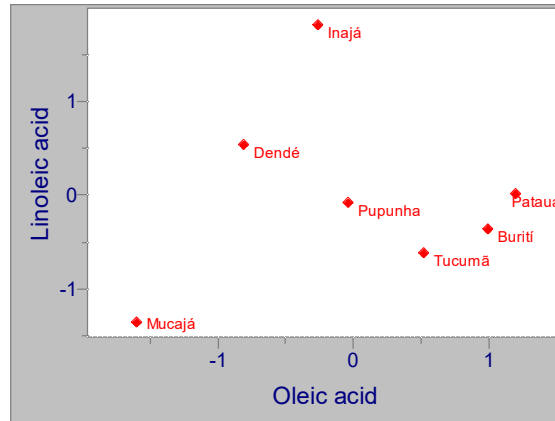
The next table contains the autoscaled [Table 4.3](#) data. When data has been autoscaled, it is sometimes said that variables have been standardized.

Table 4.5
Autoscaled Data

	Oleic Acid	Linoleic Acid
Tucumã	0.52344	-0.60820
Dendé	-0.81080	0.54619
Inajá	-0.26540	1.83025
Buriti	0.99536	-0.35920
Patauá	1.19967	0.01940
Pupunha	-0.04010	-0.06900
Mucajá	-1.60200	-1.35930

Compare [Figure 4.31](#) and [Figure 4.33](#) to the following figure, which contains autoscaled data.

Figure 4.33
Autoscaled data



RANGE SCALE

Range scaling is commonly used to prepare data points for graphing, as is the case for Pirouette’s 2D and 3D scatter plots. Each axis in such a plot is adjusted so that the data fill the plot window. This is expressed mathematically as:

$$x_{ij(rs)} = \frac{x_{ij} - x_{j(min)}}{x_{j(max)} - x_{j(min)}} \quad [4.12]$$

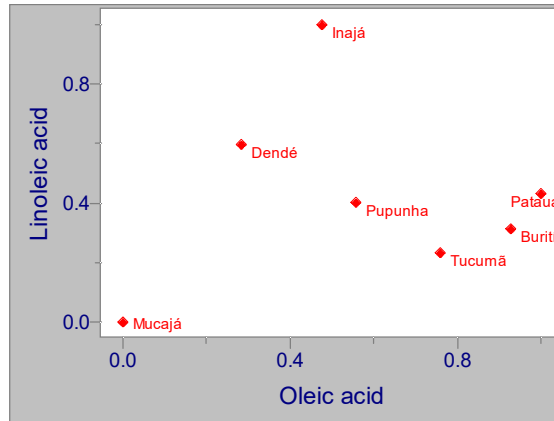
Range scaling constrains the range of values for all variables to fall between 0 and 1, inclusively. The result of range scaling the data in [Table 4.3](#) is shown below.

Table 4.6
Range Scaled Data

	Oleic Acid	Linoleic Acid
Tucumã	0.75864	0.23548
Dendé	0.28239	0.59742
Inajá	0.47708	1.00000
Burití	0.92708	0.31355
Patauá	1.00000	0.43226
Pupunha	0.55748	0.40452
Mucajá	0.00000	0.00000

Because the extreme values of each variable determine the range, this type of scaling is sensitive to the presence of outliers. The range scaled data is plotted below.

Figure 4.34
Range scaled data



PARETO SCALE

Autoscaling attempts to compensate for different magnitudes in variables but runs the risk of amplifying noise variables. A compromise between mean centering and autoscaling is a technique known as *Pareto* scaling in which the divisor is the square root of the standard deviation:

$$x_{ij(ps)} = \frac{x_{ij} - \bar{x}_j}{\sqrt{s_j}} \quad [4.13]$$

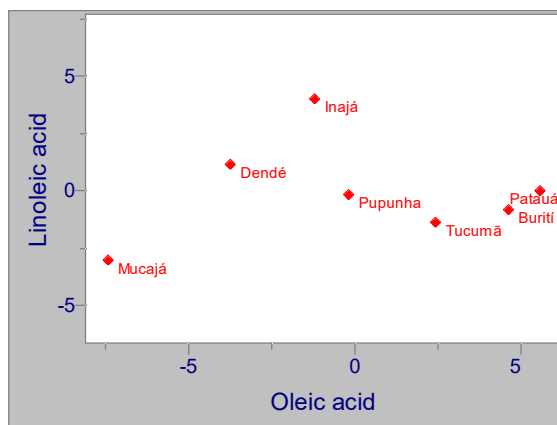
The next table contains the Pareto-scaled [Table 4.3](#) data.

Table 4.7
Pareto-scaled Data

	Oleic Acid	Linoleic Acid
Tucumã	2.426360	-1.340794
Dendé	-3.758654	1.204058
Inajá	-1.230284	4.034695
Burití	4.613874	-0.791905
Patauá	5.560934	0.042771
Pupunha	-0.186145	-0.152289
Mucajá	-7.426084	-2.996536

In the following figure, oleic acid still appears as the dominant variable, however, the spread along the linoleic acid axis is almost of the same magnitude. As a result, both variables will participate in the analysis; neither variable will be overemphasized.

Figure 4.35
Pareto scaled data



SETTING PREPROCESSING OPTIONS

Each Pirouette algorithm has its own set of options of which preprocessing is one. To set Preprocessing options, refer to [Figure 4.36](#) and use the following procedure:

- Choose the Process/Run menu item or ribbon button
- Select an Algorithm
- Select the desired Preprocessing
- Select an Exclusion set on which to run the algorithm
- Click on Add to add that Algorithm and Exclusion set configuration
- Click on Run when you have finished adding configurations

Figure 4.36
Setting
preprocessing
options

Run Configure

Algorithm:	Exclusion Sets:	Configured Runs:
HCA	Full Data	Training set PCA
PCA	Training set	Training set PLS
KNN		
SIMCA		
PLS		
PCR		
CLS		
MCR		
ALS		
PLS-DA		
LWR		
XFORM		

Algorithm Options:	Transforms:
<p>Partial Least Squares Regression</p> <p>Preprocessing: Mean-center</p> <p>Maximum Factors: Mean-center</p> <p>Validation Options: Range scale</p> <p>Validation Method: Variance scale</p> <p>Leave-out #: 1 (1-24)</p>	<p>Available: Sel</p> <p>1st Derivative</p> <p>2nd Derivative</p> <p>Align</p> <p>Baseline Correct</p> <p>Divide By >></p> <p>Log10</p> <p>MSC</p> <p>Multiply <<</p> <p>Normalize</p> <p>Smooth</p> <p>SNV</p>

When investigating a new data set, you may not know which form of preprocessing is most appropriate and should take advantage of the Run Configuration batch capability by applying several preprocessing options to a data subset and comparing the results. However, there are some rules of thumb:

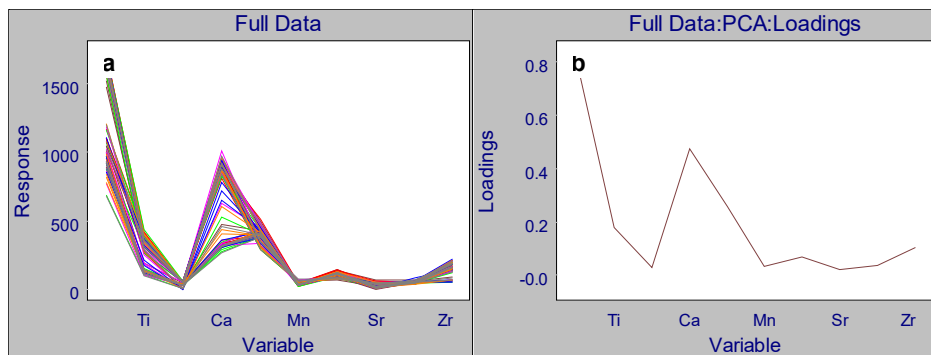
- For spectroscopic data, mean-center
- For NMR or chromatographic profile data, pareto scale
- For chromatographic peak data, usually autoscale
- For data of discrete, uncorrelated physical measurements, autoscale

PREPROCESSING AND OUTLIERS

Preprocessing increases the influence of outliers. Range scaling is worst with autoscaling a close second. This explains the importance of outlier diagnostics in variance-based algorithms.

When no preprocessing is applied, the first loading vector in PCA, PCR, PLS and SIM-CA represents the mean of the independent variable block (scaled to unit length). Thus, mean-centering or autoscaling reduces the complexity of a factor-based model by one factor. The following figure illustrates this concept; the first loading (see “Mathematical Background” in Chapter 5) has the same shape as the mean of the data.

Figure 4.37
(a) Raw data (b) First loading with no preprocessing



Calibration Transfer

Difficulties may arise when a multivariate calibration model is created with independent variable data from one instrument and applied to data collected on another instrument. Seemingly minor between-instrument differences (*e.g.*, in chromatographic stationary phase, mass spectral tuning, or wavelength registration) can reduce model reliability significantly. Two courses of action exist: either make a new model with data from the second instrument only or apply a computational adjustment to mitigate the instrument differences. The second approach, known as calibration transfer, has been treated extensively in the chemometric literature⁷⁻⁸. Pirouette implements a form of calibration transfer related to Procrustes Analysis⁹. It requires collecting profiles of matching samples on both instruments from which an intermediate matrix is calculated. This matrix adjusts profiles from the second instrument to look like those collected on the first. Thus, it can be thought of as a data preprocessing step which occurs after acquisition of the data but before application of the model.

Ideally, the matching profiles would come from the identical sample run on both instruments, but in reality this may not be practical. Instead it may be necessary to find or create new samples of like composition. For regression models, a matching profile is defined by identical dependent variable settings; for classification models, the assigned category must be identical.

Note: *When processing raw chromatographic data, the primary impact on calibration transfer is the variation in retention time. In many cases, this time axis instability can be corrected using another Infometrix software product, LineUp™. This is discussed more completely in the LineUp User Guide.*

SUBSET SELECTION

A crucial first step is choosing the so-called transfer or repeat samples, those training set samples whose matching profiles will also be acquired on the second instrument. A successful calibration transfer requires that the most representative profiles in the training set be stored in the model when it is created. Pirouette uses the maximum distance method of Kennard and Stone¹⁰ to find these profiles. Enough transfer samples should be stored in the model to adequately represent the population modeled. Pirouette stores data for up to 10 samples for a regression model and 10 samples per category for a classification model. See “[Calibration Transfer](#)” on page 6-29 and “[Calibration Transfer](#)” on page 7-56 for important details about the implementation for regression and classification algorithms, respectively.

Four types of adjustments are possible in Pirouette: additive, multiplicative, direct and piecewise. The user specifies the type via the prediction preferences; see “[Prediction](#)” on page 10-19. The mathematical foundation of each follows.

ADDITIVE AND MULTIPLICATIVE ADJUSTMENT

Often response differences in the training and prediction profiles are due to a predictable but unknown offset. This difference can be accounted for by an additive or a multiplicative correction. In the additive correction, the difference between the mean spectrum of the training set transfer samples and the mean spectrum of the matching prediction transfer samples is computed. The difference is then added to all prediction spectra before the model is applied. Thus, the prediction profiles are simply modified in such a way that the means between the training and predictions sets are coincident. Similarly, in the multiplicative correction, all prediction spectra are multiplied by the ratio of the two means. If the differences in the instrument responses vary with signal or are more complex than a simple additive or multiplicative bias, neither approach will be as reliable as the corrections discussed below.

DIRECT AND PIECEWISE ADJUSTMENT

If \mathbf{X}_1 is the matrix containing the transfer sample profiles from one instrument and \mathbf{X}_2 contains the matching profiles from another instrument (or from the first instrument after acquisition conditions have changed), then we seek a transformation matrix \mathbf{F} such that:

$$\mathbf{X}_1 = \mathbf{X}_2\mathbf{F} + \mathbf{E} \quad [4.14]$$

so

$$\mathbf{F} = \mathbf{X}_2^{-1}\mathbf{X}_1 \quad [4.15]$$

Since \mathbf{X}_2 is unlikely to be invertible, SVD is used (see “Principal Component Regression” on page 7-4) to form the inverse:

$$\mathbf{X}_2 = \mathbf{U}\mathbf{S}\mathbf{V}^T \text{ and } \mathbf{X}_2^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T \quad [4.16]$$

A translational correction \mathbf{b} completes the transformation. If $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ are the respective means of the data subsets, then:

$$\mathbf{b} = \bar{\mathbf{x}}_1 - \mathbf{F}^T\bar{\mathbf{x}}_2 \quad [4.17]$$

Together, \mathbf{F} and \mathbf{b} specify the transfer of calibration adjustment. The adjusted prediction profile \mathbf{x}_{2a} (from the second instrument) is then:

$$\mathbf{x}_{2a} = \mathbf{F}^T\mathbf{x}_{2u} + \mathbf{b} \quad [4.18]$$

where \mathbf{x}_{2u} is the original (unadjusted) prediction profile--the unknown.

This form of adjustment, incorporating a rotation, a stretching and a translation, can accommodate most sources of between-instrument differences. It is often referred to as Direct Standardization (DS). In DS, all variables in \mathbf{X}_2 are related to each variable in \mathbf{X}_1 to form the transformation matrix \mathbf{F} .

It is also possible to relate a small number of variables (*i.e.*, a window) in \mathbf{X}_2 to a single variable in \mathbf{X}_1 . The transform matrix is then composed by incorporating, for each variable in \mathbf{X}_1 , the regression vector from this window-to-variable relationship. This approach, often referred to as Piecewise Direct Standardization (PDS)⁷, PDS transfer, requires the specification of a window size. Reports in the literature suggest that the window size be at least as large as the chemical rank of the data in the subset.

Final Remarks

What remains now is to establish the relationship between the contents of the data file discussed in this chapter and the \mathbf{X} matrix, the \mathbf{x} block on which each algorithm operates. This requires introducing some nomenclature and specifying a few niggling details. The nomenclature is necessary for the algorithm discussions in the following chapters. For many users the details are of no importance. However, we supply them because some commercial software packages perform undocumented computations which are only discovered when one attempts to duplicate the results using a different package. When there is no universally accepted *right way* (only a variety of plausible ways) to implement a software solution, it is not surprising that different packages take different approaches. Our intent is merely to document the choices imbedded in Pirouette and point out where these choices might impact the user.

Pirouette assumes that samples are stored as rows in a matrix. So a data file containing information about n samples must contain n rows. Pirouette recognizes three kinds of variables: x , y , and c (for class). Statisticians refer to x and y variables as independent and dependent, respectively. The number of columns in a Pirouette data file is equal to

4 Preparing for Analysis: References

the sum of the number of x, y and c variables. Variables are assumed to be independent until the user specifies a different type.

Note: *Pirouette does not count the row containing variable names and the column containing sample names in a data file and differs from traditional spreadsheets in this way.*

Some differences between transforms and preprocessing have already been mentioned but others are now pertinent. It is possible to create an independent object containing the transformed data, while preprocessed data are computed and stored for each algorithm run. On the other hand, the preprocessed data are made available after a run as one of the computed objects. Transforms operate on x variables only, while preprocessing operates on x and y variables. For any given subset of a full data file, transforms are applied first, then the transformed data are preprocessed, then an algorithm is run. So it is the transformed and preprocessed x variables which comprise the \mathbf{X} matrix used in computations. The symbol \mathbf{X}_{raw} designates untransformed and unpreprocessed x variables, the symbol $\mathbf{X}_{\text{trans}}$ will designate transformed x variables.

In matrix notation, a sample is a single row vector of data, \mathbf{x} , which can be written as:

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_m] \quad [4.19]$$

where elements of the row vector are the m measurements (*i.e.*, the independent variables) made on the sample. When n samples are available, each vector is stacked with the others to form a matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad [4.20]$$

where the element x_{ij} is the j^{th} variable measurement on the i^{th} sample, and the elements of \mathbf{X} may have been transformed and/or preprocessed. Throughout the next chapters, quantities which occur repeatedly are assigned the same symbol: n = the number of samples, m = the number of independent variables.

For most Pirouette computations (that is, in PCA, SIMCA, PLS-DA, PCR, PLS, CLS, ALS, LWR, MCR), quantities called X Residuals are computed. They represent the difference between the independent variables before and after some fitting procedure. An obvious question is: *which* independent variables are meant: \mathbf{X}_{raw} , $\mathbf{X}_{\text{trans}}$ or \mathbf{X} ? In Pirouette, X residuals correspond to $\mathbf{X}_{\text{trans}}$. This distinction, which becomes important when data are autoscaled, range scaled or variance scaled, is discussed later in the context of each algorithm.

References

1. Savitzky, A. and Golay, M.J.E.; "Smoothing and Differentiation of Data by Simplified Least Squares Procedures" *Anal. Chem.* 36:1627-1639 (1964).

2. Gorry, P.A.; "General Least-Squares Smoothing and Differentiation by the Convolution (Savitzky-Golay) Method" *Anal. Chem.* 62:570-573 (1990).
3. Johansson, E.; Wold, S. and Sjodin, K.; "Minimizing Effects of Closure on Analytical Data." *Anal. Chem.* 56: 1685-1688 (1984).
4. Kováts, E., *Adv. Chromatogr.*, 1: 229 (1965).
5. Göbller, A., *J. Chromatogr. Sci.*, 9: 311 (1972).
6. Seasholtz, M.B. and Kowalski, B.R.; "The Effect of Mean Centering on Prediction in Multivariate Calibration." *J. Chemometrics* 6: 103-111 (1991).
7. Wang, Y. and Kowalski, B.R.; "Multivariate instrument standardization." *Anal. Chem.*, (1991) 63: 2750-2756.
8. Bouveresse, E. and Massart, D.L.; "Standardization of Near_Infrared Spectrometric Instruments: A Review." *Vibrational Spectroscopy* 11: 3-15 (1996).
9. Anderson, C.E. and Kalivas, J.H.; "Fundamentals of Calibration Transfer through Procrustes Analysis." *Appl. Spec.*, (1999) 53: 1268-1276.
10. Kennard and Stone; *Technometrics*, (1969) 11: 137.

Exploratory Analysis

Contents

Hierarchical Cluster Analysis	5-1
Principal Component Analysis	5-13
References	5-46

In many technical fields, large quantities of multivariate data are available. The goal of exploratory analysis is to provide a quality check on the data, determine its information content and pinpoint key measurements. It may also establish the viability of the data set with regard to regression or classification model-building.

Exploratory analysis is the computation and graphical display of patterns of association in the independent variables (*i.e.*, the x block) of multivariate data sets. Exploratory algorithms reduce large and complex data sets to a suite of best views; these views provide insight into both the x block structure and correlations existing among samples and/or independent variables. Pirouette implements two exploratory techniques: Hierarchical Cluster Analysis and Principal Component Analysis.

Hierarchical Cluster Analysis

In Hierarchical Cluster Analysis (HCA), distances between pairs of samples (or variables) are calculated and compared. When distances between samples are relatively small, this implies that the samples are similar, at least with respect to the measurements in hand. Dissimilar samples will be separated by relatively large distances. Known in biological sciences as numerical taxonomy, HCA groups data into clusters having similar attributes.

The primary purpose of HCA is to present data in a manner which emphasizes natural groupings. In contrast to techniques that group samples into pre-existing categories, HCA seeks to define those categories in the first place. The presentation of HCA results in the form of a dendrogram facilitates the visual recognition of such categories. HCA can focus on samples or variables. Clustering of samples reveals similarities among the samples while clustering of variables pinpoints intervariable relationships. Although the remarks in this chapter are couched in terms of sample clustering, they apply equally well to variable clustering.

Clustering in Pirouette is of an agglomerative type. In such an approach, we start with each sample defined as its own cluster, then begin grouping samples together to form new clusters until all samples are part of a single cluster. In contrast with the agglomer-

ative methods, divisive methods start with a single cluster composed of all samples, then divide clusters until each sample becomes a cluster unto itself. There are, of course, situations in which one of these approaches will outperform the other (*c.f.*, Hartigan¹).

MATHEMATICAL BACKGROUND

This section introduces the mathematical foundations of the HCA algorithm and defines several important terms.

Distance Measures

Multivariate distance is computed on the independent variable block, which is the transformed and preprocessed matrix \mathbf{X} . The multivariate distance d_{ab} between two sample vectors, \mathbf{a} and \mathbf{b} , is determined by computing differences at each of the m variables:

$$d_{ab} = \left[\sum_j^m (x_{aj} - x_{bj})^M \right]^{1/M} \quad [5.1]$$

where M is the order of the distance. The distance d_{ab} is sometimes called a Minkowski distance.

The City Block or Manhattan Distance, where $M = 1$, finds use mostly with categorical data types:

$$d_{ab} = \sum_j^m |x_{aj} - x_{bj}| \quad [5.2]$$

The most common metric in multivariate analysis, and the one featured in Pirouette, is Euclidean distance, where $M = 2$:

$$d_{ab} = \left[\sum_j^m (x_{aj} - x_{bj})^2 \right]^{1/2} \quad [5.3]$$

Other distances, where $M > 2$, are less commonly encountered.

Similarity

Because inter-sample distances vary with the type and number of measurements, it is customary to transform them onto a somewhat more standard scale of similarity:

$$similarity_{ab} = 1 - \frac{d_{ab}}{d_{max}} \quad [5.4]$$

where d_{max} is the largest distance in the data set. On this scale, a value of 1 is assigned to identical samples and a value of 0 to the most dissimilar samples.

Linkage Method Definitions

After distances between all pairs of samples have been computed, the two most similar samples are linked. Once a cluster is linked to another cluster, they form a single new

cluster. After distances between this new cluster and all other existing clusters are determined, the smallest inter-cluster distance is again sought and another linkage formed. This process continues until all samples/clusters are linked.

Pirouette’s several approaches to establishing links between samples/clusters are defined below and discussed in detail in “Linkage Methods Illustrated” on page 5-5. The distance between a newly formed cluster A—B and a previously existing cluster C is calculated via one of the formulas below where n_i is the number of samples in cluster i.

Single Link

$$d_{AB \Rightarrow C} = 0.5d_{AC} + 0.5d_{BC} - 0.5|d_{AC} - d_{BC}| \quad [5.5]$$

Complete Link

$$d_{AB \Rightarrow C} = 0.5d_{AC} + 0.5d_{BC} + 0.5|d_{AC} - d_{BC}| \quad [5.6]$$

Centroid Link

$$d_{AB \Rightarrow C} = \left(\frac{n_A d_{AC}^2}{n_A + n_B} + \frac{n_B d_{BC}^2}{n_A + n_B} - \frac{n_A n_B d_{AB}^2}{(n_A + n_B)^2} \right)^{1/2} \quad [5.7]$$

Incremental Link

$$d_{AB \Rightarrow C} = \left(\frac{(n_A + n_C)d_{AC}^2 + (n_B + n_C)d_{BC}^2 - n_C d_{AB}^2}{n_A + n_B + n_C} \right)^{1/2} \quad [5.8]$$

Median Link

$$d_{AB \Rightarrow C} = (0.5d_{AC}^2 + 0.5d_{BC}^2 - 0.25d_{AB}^2)^{1/2} \quad [5.9]$$

Group Average Link

$$d_{AB \Rightarrow C} = \frac{n_A d_{AC} + n_B d_{BC}}{n_A + n_B} \quad [5.10]$$

Flexible Link

$$d_{AB \Rightarrow C} = (0.625d_{AC}^2 + 0.625d_{BC}^2 - 0.25d_{AB}^2)^{1/2} \quad [5.11]$$

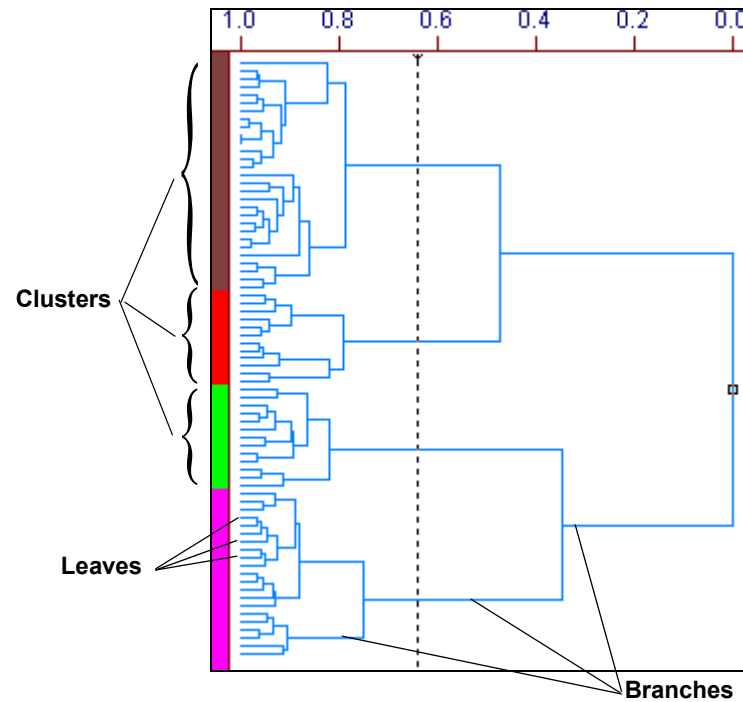
HCA Options

Some data sources are able to generate a square matrix of similarities or distances, obviating the need for Pirouette to perform that portion of the calculations. To tell Pirouette to skip past the preliminary distance calculations, choose “Euclidean (no init)” for the Distance Metric. The algorithm will then immediately proceed to the designated linkage step as described above.

HCA OBJECTS

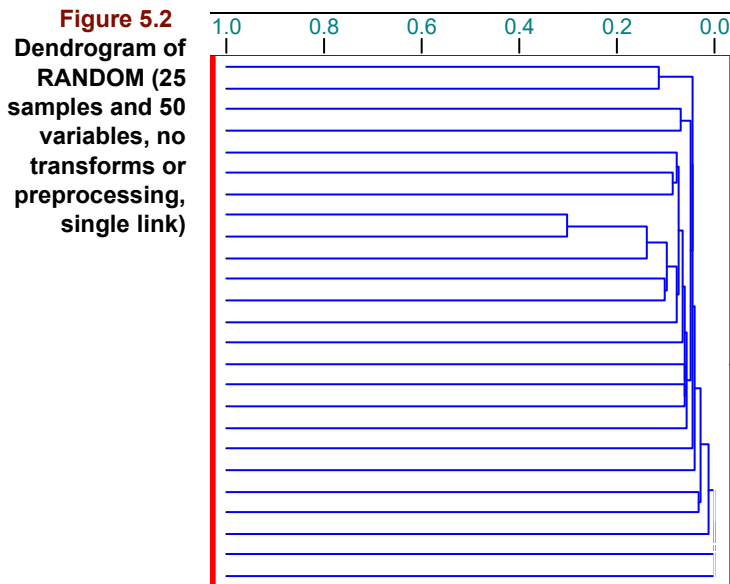
Dealing with distances or similarity values as numbers in a table is a daunting task. Graphical representation of clustering results is critical to effective HCA interpretation. Pirouette employs a graph form called a dendrogram to depict the similarity of samples or variables. It is a tree-shaped map constructed from the table of distances. Branch lengths are proportional to the distances between linked clusters. The shape of the dendrogram (*i.e.*, the connectivity of the various branches) is a function of the linkage method. A dendrogram with 75 samples is shown below.

Figure 5.1
Dendrogram with
four clusters at a
similarity value of 0.5



The terminus of the branches on the far left of the dendrogram, called leaves, represent single samples. The length of the branches linking two clusters is related to their similarity. The longer the branch, the less the similarity; the shorter the branch, the greater the similarity and, therefore, the smaller the intercluster distance. Similarity is plotted along the top of the graphic with 1.0 corresponding to an exact duplicate and 0.0 indicating maximum distance and dissimilarity. The dotted vertical line slices through the dendrogram in [Figure 5.1](#) at a similarity value of approximately 0.65, where four clusters can be distinguished. The lengths of these four branches are long compared to the branches connecting clusters to the left. If noise were added to the data, the leaf lengths would grow and the branch lengths shorten. In random data, leaves are often as long as or longer than most of the branches.

Note: *Even random numbers produce clusters. To familiarize yourself with the look of a dendrogram constructed from random data, open the file shipped with Pirouette called RAN-DOM.XLS. The result of HCA on this data set is shown below.*

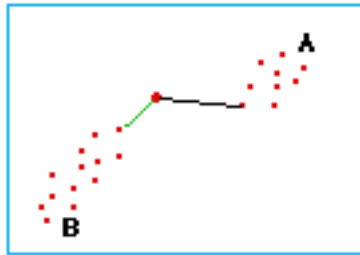


LINKAGE METHODS ILLUSTRATED

Whenever HCA is run, an approach to deciding how samples are grouped must be chosen. There are three general types of linkage methods: nearest neighbor, farthest neighbor and centroidal.

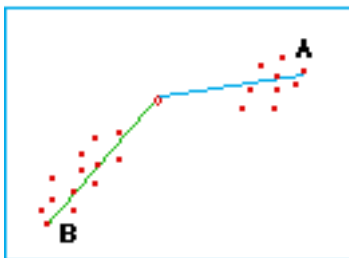
The simplest is nearest neighbor linking, which is based strictly on the distance from any one sample (alone or in a cluster) to its nearest neighbor. The sample is assigned to the cluster containing that nearest neighbor. This concept is depicted below; the unknown (big circle) is closer to cluster B's closest member than to cluster A's closest member.

Figure 5.3
Nearest neighbor
linking



The farthest neighbor method assigns a sample to the cluster whose farthest neighbor is closest to that sample. This concept is depicted in the next figure where the unknown is closer to the farthest member of cluster A than to the farthest member of cluster B.

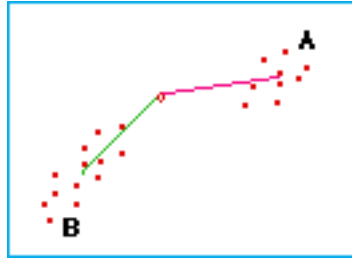
Figure 5.4
Farthest neighbor
linking



5 Exploratory Analysis: Hierarchical Cluster Analysis

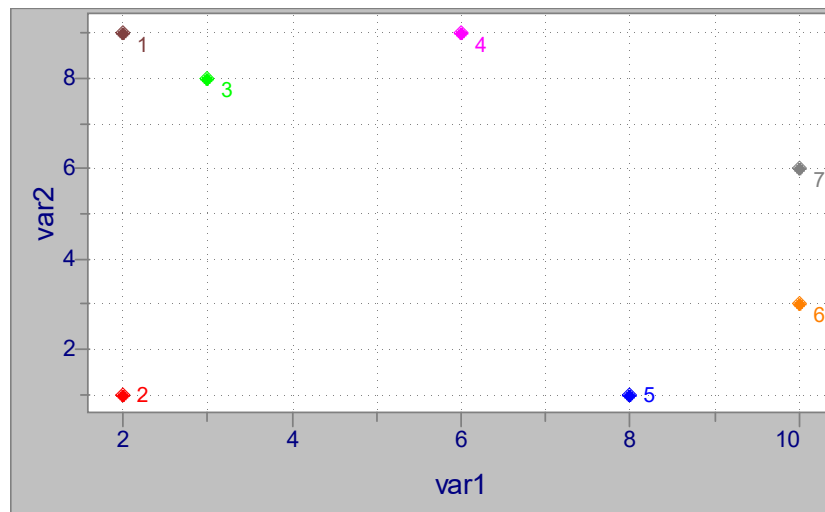
In contrast to nearest and farthest neighbor linking, centroidal linking assigns a sample to the cluster whose center is nearest. The basic concept is depicted in below where the unknown is closer to the center of cluster A than to the center of cluster B.

Figure 5.5
Centroidal linking



The remainder of this discussion contrasts Pirouette's various linkage methods using data from a file called SEVEN.DAT. A scatter plot of SEVEN, which contains seven samples and two variables, is shown below.

Figure 5.6
Scatter plot of
samples in SEVEN



The intersample distances for these data are shown in the following table.

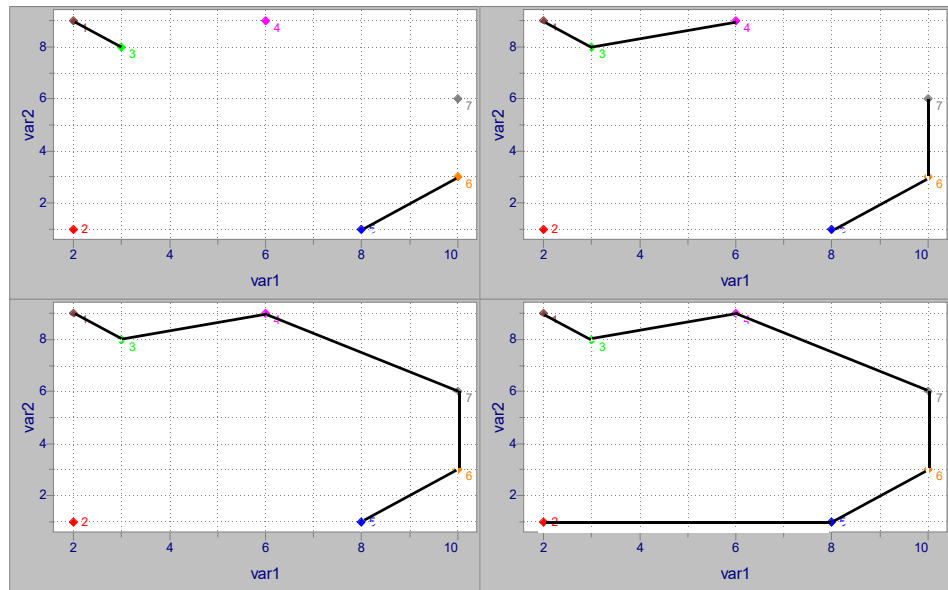
Table 5.1
Distances between
samples in SEVEN

Sample #	1	2	3	4	5	6	7
1	0	8.0	1.4	4.0	10.0	10.0	8.5
2		0	7.1	8.9	6.0	8.2	9.4
3			0	3.2	8.6	8.6	7.3
4				0	8.2	7.2	5.0
5					0	2.8	5.4
6						0	3.0
7							0

Nearest Neighbor Linkage

Pirouette calls its nearest neighbor linking method Single Link. The following graphic illustrates how Single Link linking is applied to the SEVEN data.

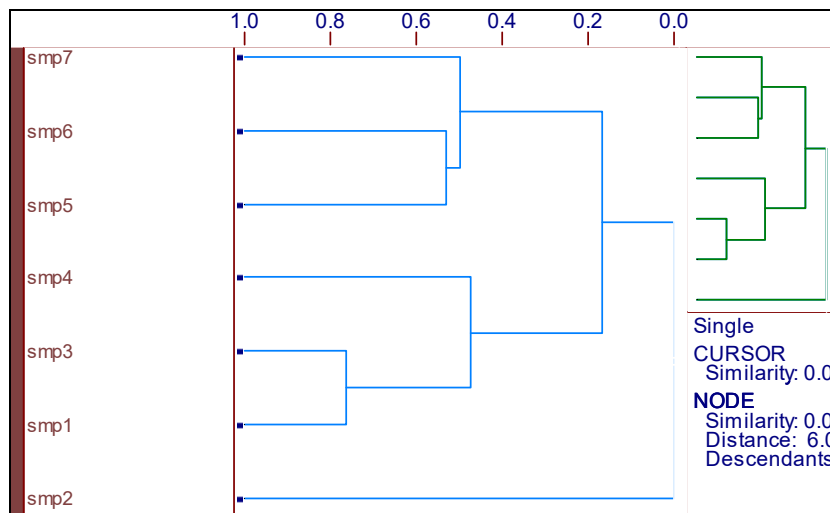
Figure 5.7
Succession of linkages with Single Link



Because samples 1 and 3 have the smallest intersample distance, as shown in Table 5.1, they are joined first. The second link is between samples 5 and 6. This sequence is depicted in Figure 5.7a. Linking of the samples proceeds from top left to top right, bottom left to bottom right.

Figure 5.8 shows this same set of linkages in the dendrogram view. Bigger clusters form as samples are successively linked. Because samples 1 and 3 are most similar, they have the shortest branches. This group of two is next connected to sample 4. Samples 5, 6 and 7 join in a similar fashion. The two clusters of three samples are then connected. Sample 2 has the longest branch because, being most dissimilar, it is linked last when Single Link is applied.

Figure 5.8
Single Link dendrogram of SEVEN data



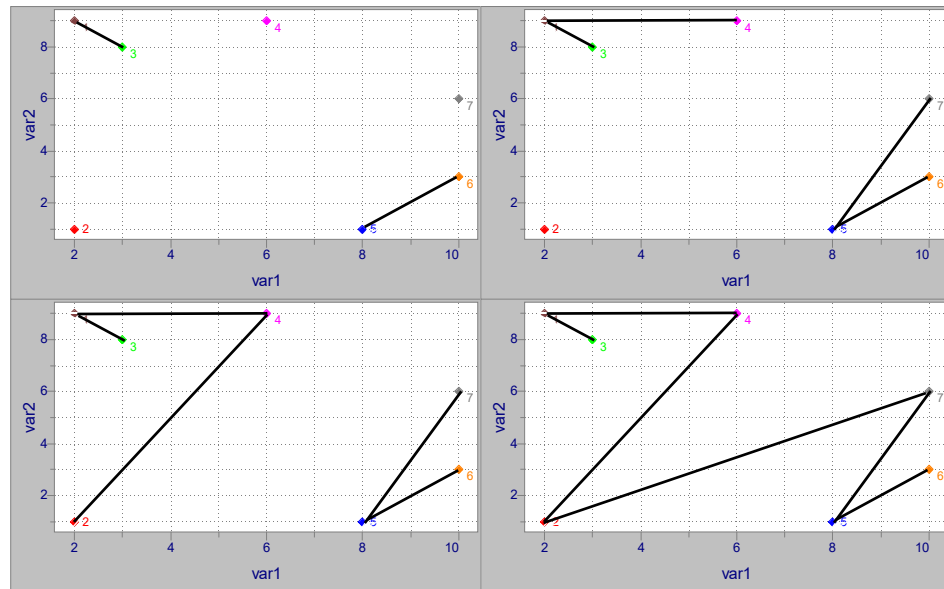
Farthest Neighbor Linkage

Pirouette's farthest neighbor linking method is called Complete Link. Figure 5.9 illustrates Complete Link clustering applied to the SEVEN data. With Complete Link, sample

5 Exploratory Analysis: Hierarchical Cluster Analysis

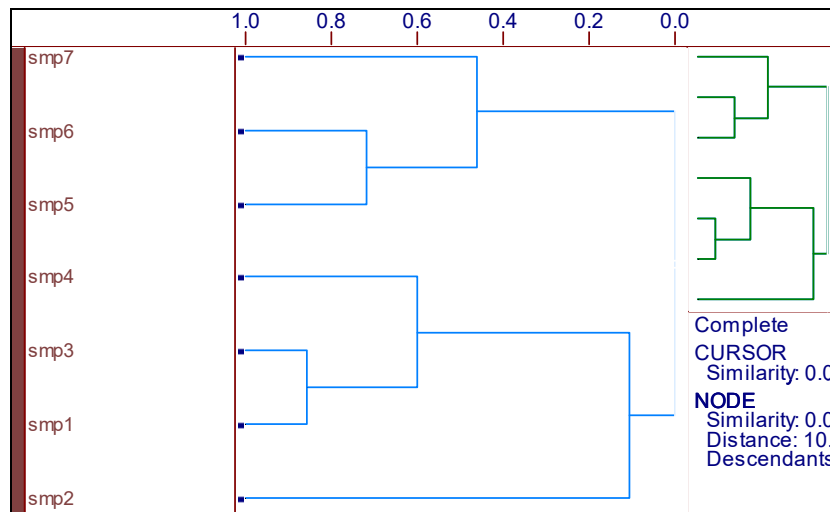
2 is first grouped with the 1-3-4 cluster. However, with Single Link, sample 2 did not link with any cluster until the last step.

Figure 5.9
Succession of linkages with Complete Link



The dendrogram below shows that sample 2 is more similar to cluster 1-3-4 than it is to cluster 5-6-7 when Complete Link is applied.

Figure 5.10
Complete Link dendrogram of SEVEN data



Centroidal Linkage

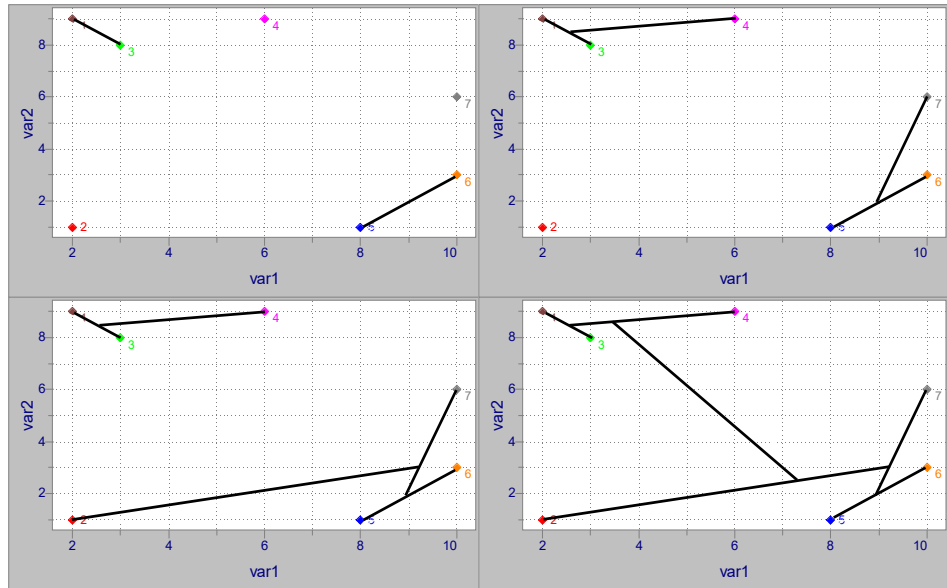
Five different centroidal linkage methods are implemented in Pirouette. All attempt to find a central region of a cluster from which to construct distances; they differ in the way that the cluster center is computed.

- **Centroid Link** uses the average position in a cluster
- **Median Link** finds the median point in a cluster
- **Flexible Link** is a weighted Median Link

- **Group Average Link** is a Centroid Link variant minimizing dendrogram crossovers (see below for details)
- **Incremental Link** employs a sum of squares approach in calculating intercluster distances

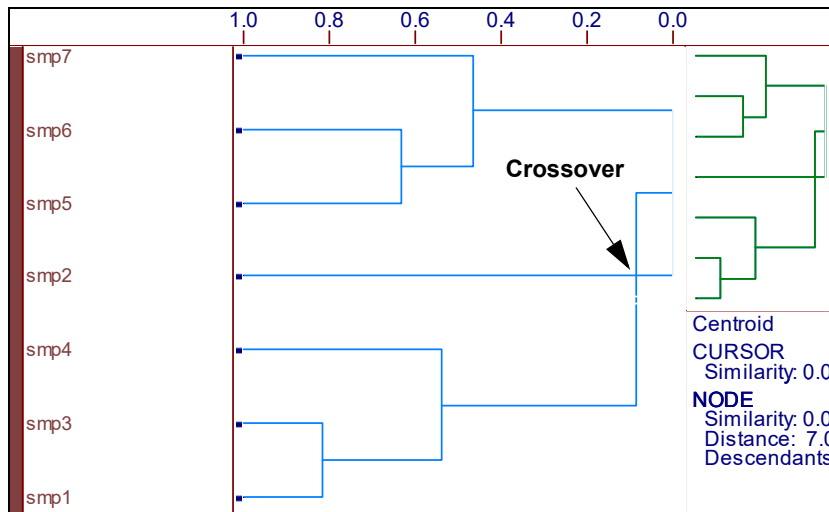
The next graphic illustrates Centroid Link as applied to the SEVEN data.

Figure 5.11
Succession of linkages with Centroid Link



Notice that Centroid Link initially groups the samples into the same clusters as Single Link but depicts sample 2 as being more similar to the 5-6-7 cluster than was true for Single Link (*i.e.*, it links sample 2 to the 5-6-7 cluster before linking cluster 1-3-4). Centroid Link occasionally inverts a node as shown in Figure 5.12; the inversion is referred to as a crossover. Thus, although sample 2 links to the 5-6-7 group, that intercluster distance is longer than the distance between cluster 1-3-4 and cluster 2-5-6-7. Trace out the linking route in Figure 5.11 to confirm the structure of the dendrogram shown below.

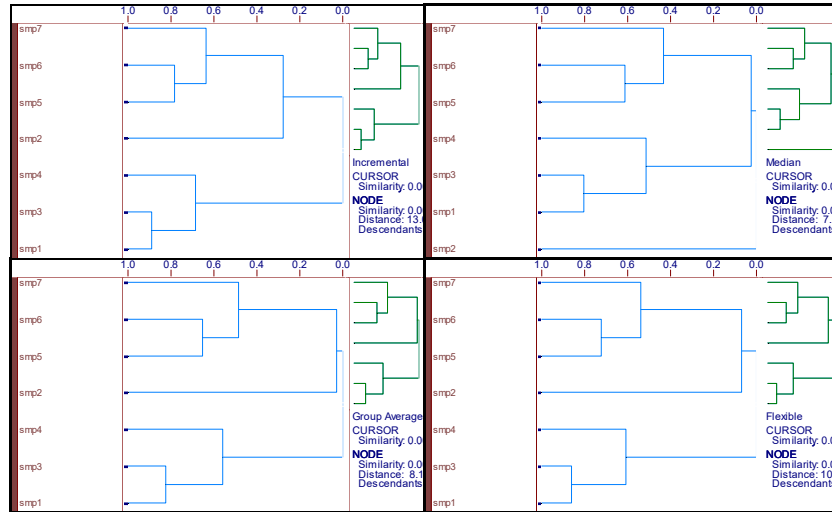
Figure 5.12
Centroid Link dendrogram of SEVEN data with a crossover



5 Exploratory Analysis: Hierarchical Cluster Analysis

Figure 5.13 shows the results of the other centroidal methods applied to the SEVEN data. Median Link gives the same dendrogram structure as Single Link. Incremental Link, Group Average Link, and Flexible Link resemble Centroidal Link but do not suffer from crossover.

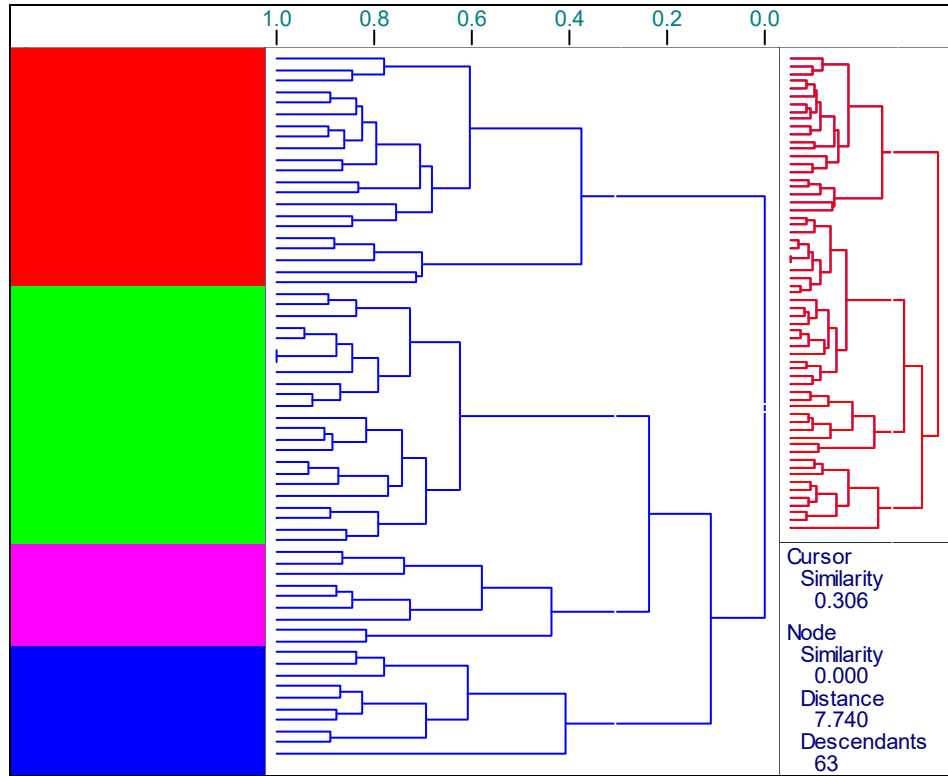
Figure 5.13
Dendrograms of SEVEN data with Incremental, Median, Group Average, and Flexible Links



CHOOSING A LINKAGE METHOD

Choosing a linkage method is both an art and a science. If clusters are distinct, dendrogram shape is hardly affected by the choice. For example, the ARCH dendrogram in the next figure has such tight clusters that comparable results would be expected from any linkage method.

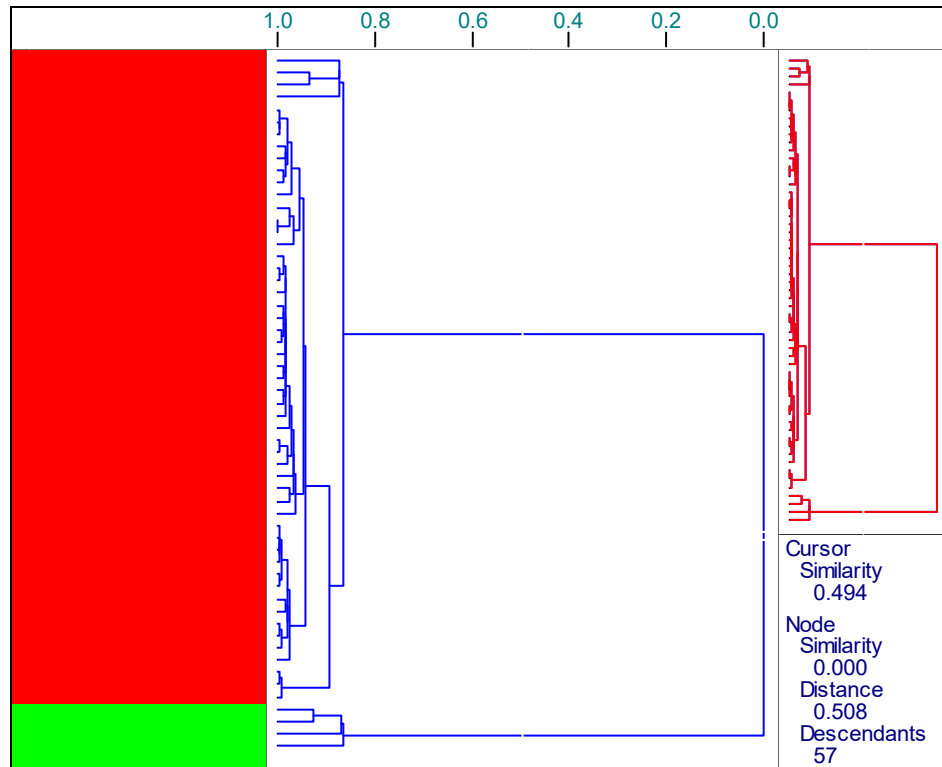
Figure 5.14
Distinct clusters in
the ARCH data



However, small clusters can produce misleading dendrograms if either a farthest neighbor (*i.e.*, Complete Link) or a centroidal method is applied. In these cases Single Link is usually the better choice. As shown in the following figure, OCTANE20 data forms three groups, two having only four samples each.

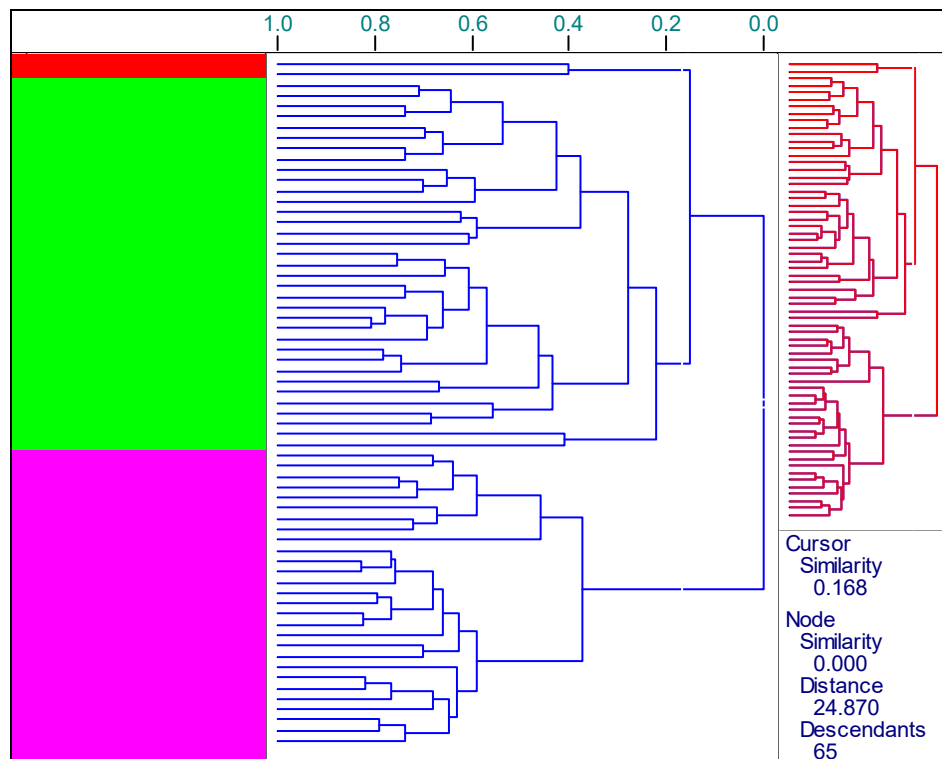
5 Exploratory Analysis: Hierarchical Cluster Analysis

Figure 5.15
Small clusters in the
OCTANE20 data



Incremental Link works better than other methods in instances where two groups of samples differ only slightly, as shown next in the ALCOHOL dendrogram.

Figure 5.16
Poorly separated
clusters in the
ALCOHOL data



The appropriateness of a linkage method depends on not only the particulars of the data set but also the purpose of the analysis. If you intend to create a KNN classification model, consider Single Link, which yields a view consistent with KNN results. It is good practice to try both Single Link and Complete Link initially. If they show clusters composed of roughly the same samples, go no further. Where cluster memberships differ dramatically, centroidal methods should be investigated to see if they favor Single Link or Complete Link results. In cases where centroidal methods give noticeably different cluster assignments from both Single Link and Complete Link, more extensive comparisons (e.g., with PCA scores plots discussed in the next section) are warranted.

Principal Component Analysis

Principal Component Analysis (PCA) is a powerful visualization tool and thus finds use in exploratory analysis. Like HCA, it can represent graphically intersample and intervariable relationships. Moreover, it provides a way to reduce the effective dimensionality of the data. PCA finds linear combinations of the original independent variables which account for maximal amounts of variation. The concept of variation is as central to PCA as multivariate distance is to HCA and requires some elaboration.

GENERAL CONCEPTS

Variance vs. Variability

For any vector \mathbf{x} containing n elements, the variation over the vector might be generally defined as simply the sum of squares:

$$\sum_i^n x_i^2 \quad [5.12]$$

This should be contrasted with the statistical variance, s^2 , which is defined as:

$$s^2 = \frac{1}{n-1} \sum_i^n (x_i - \bar{x})^2 \quad [5.13]$$

The standard deviation, s , the square root of the variance, is perhaps a more familiar measure of spread in a series of values. The variation/variance distinction is moot whenever preprocessing options that subtract the mean of each variable are used; the two quantities differ only by a factor of $n-1$. Otherwise, you should be aware of the variation/variance distinction, particularly when comparing Pirouette results with other commercial statistical software packages. In the discussion which follows, the term *variance* is often employed to describe the quantity computed in [equation 5.12](#). This conflation of variation and variance is reasonable to non-statisticians given the prevalence of mean-centering. For a discussion of the various preprocessing options in Pirouette, see [“Preprocessing” on page 4-26](#).

In the same way that HCA is built on the assumption that small multivariate distance implies similarity, PCA is built on the assumption that variation implies information. Vari-

ation might be classified as either relevant or irrelevant. Noise is an example of irrelevant variation.

Correlation

Perhaps you are convinced that visualizing large data sets is useful but are wondering why it is advantageous to reduce their dimensionality. Multivariate data sets typically contain values produced by non-specific sensors, particularly so in chemistry. No single probe, for example, responds only to the freshness of fruit or to an individual's intelligence level. To mitigate this lack of specificity, data are collected from many sensors in the hope that each captures an aspect of a property or category under investigation. However, information acquired by sets of sensors is often partially redundant; in statistical terms, the measurement variables are correlated. If data on several samples are collected from fifty non-specific and partially redundant sensors, the observed variance pattern is probably not due to fifty independent phenomena but is instead determined by a much smaller number of factors. This number might be viewed as the intrinsic dimensionality of the data. PCA provides a way to find those factors.

Terminology

Many terms are used for the linear combinations of original independent variables found by PCA: latent variables, abstract factors, principal components, loadings and eigenvectors. The first two colorful terms evoke the psychometric origins of the method. The last term is the least ambiguous, referring to a result of a particular matrix decomposition discussed in most numerical analysis texts². PCA eigenvectors have several desirable properties. First, they are mutually orthogonal, which is the same as saying they are uncorrelated. Second, eigenvectors can be computed in order of decreasing variance. Thus, the first eigenvector accounts for the maximum amount of variance and each successive eigenvector accounts for less of the remaining variance in the data. There are as many unique eigenvectors as original variables or samples, whichever is smaller.

Note that the term *independent variable* is somewhat misleading to the non-statistician. PCA is a particularly effective multivariate technique when the original so-called independent variables are *not* independent of each other. Finding linear combinations of these variables which are uncorrelated (that is, eigenvectors) gets around the difficulties caused by intervariable correlations.

Visualization

To visually characterize a data set, we want to construct a small number of 2D or 3D plots with important variables on the axes so that key relationships among samples can be identified. Unfortunately, given the non-specificity and redundancy mentioned above, we cannot know *a priori* which variables are most important. Moreover, the number of possible combinations of axes variables grows dramatically as the number of variables increases. However, if the first two or three principal components are placed on the axes instead of original variables, the intersample relationships displayed are guaranteed to be optimal from the point of view of variance captured. This explains PCA's strength as a visualization technique.

Returning to the idea of reducing the dimensionality of a data set, it implies that the data contain irrelevant or random variation, some of which can be removed by retaining only the principal components which capture relevant variation. After the optimal number of principal components has been determined, the data set is *not* smaller than it was before PCA. None of the original variables has been removed. Instead, certain combinations of them are disregarded.

PCA provides the best possible view of variability in the independent variable block, which reveals if there is natural clustering in the data and if there are outlier samples. It may also be possible to ascribe chemical (or biological or physical) meaning to the data patterns which emerge from PCA and to estimate what portion of the measurement space is noise. Finally, a PCA model can be created and serve as a benchmark for comparisons with future samples.

Note: A general understanding of PCA is necessary because of its role in pattern recognition described above and because both SIMCA (“Soft Independent Modeling of Class Analogy” in Chapter 6) and PCR (“Factor Based Regression” in Chapter 7) are based on principal components.

A Concrete Example

If the ideas of maximum variance axes and dimensionality reduction are applied to a banana, the banana’s length axis is its first principal component (PC). A second axis, perpendicular to the first, describes the second PC: the span of its curvature. The third PC is the axis drawn at right angles to both PC1 and PC2: the fruit’s thickness. The following figure shows a banana in perspective and in three different 2D views based on the PC axes just described.

Figure 5.17
Banana views

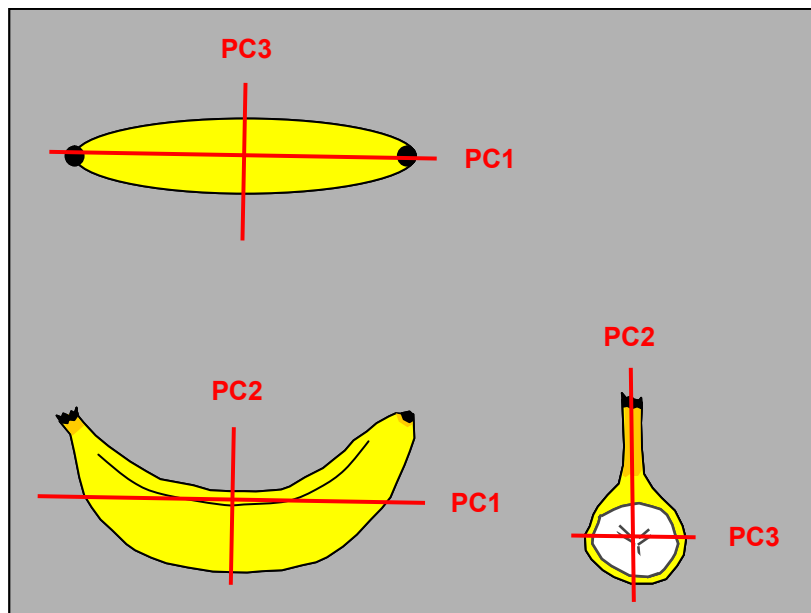
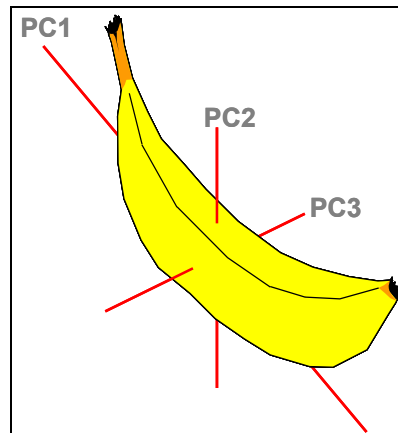


Figure 5.18 shows a perspective view with the three principal component axes superimposed. The banana shown in this figure has merely been transformed onto three new plotting axes. No reduction in dimensionality has been achieved; an object possessing x, y and z coordinates is still displayed in three dimensions. However, if it *were* necessary to reduce the 3D banana to a planar representation, the view shown in the lower left of Figure 5.17, that of PC1 vs. PC2, is the most informative.

Figure 5.18
A banana with 3 PC axes



MATHEMATICAL BACKGROUND

PCA is based on the idea of expressing a matrix \mathbf{X} (defined in the “Final Remarks” on page 4-35) as the product of two other matrices—the scores matrix \mathbf{T} and the transpose of the loadings matrix \mathbf{L} :

$$\mathbf{X} = \mathbf{T}\mathbf{L}^T \quad [5.14]$$

Note: Those unfamiliar with linear algebra might want to read [Chapter 17, An Introduction to Matrix Math](#).

The size of the matrices deserves some comment. \mathbf{X} contains n rows and m columns, corresponding to the number of samples and independent variables, respectively. The scores matrix \mathbf{T} contains n rows and g columns; the loadings matrix \mathbf{L} contains m rows and g columns; the transpose of \mathbf{L} contains g rows and m columns. The value of g is the number of independent variables or samples, whichever is smaller and is often referred to as the mathematical rank of the matrix.

If only the first k columns of the scores and loadings matrices (where k is less than g) are considered relevant and retained, then

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{T}_k \mathbf{L}_k^T \quad [5.15]$$

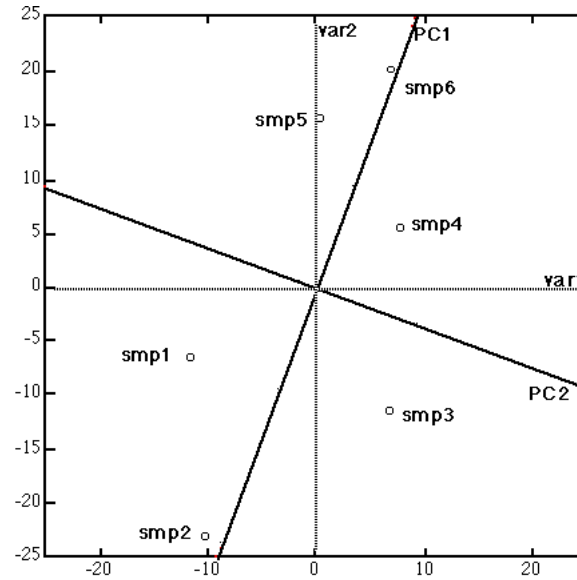
where $\hat{\mathbf{X}}$ is an estimate of \mathbf{X} , and the dimensionality of the data is said to have been reduced. This language is a bit misleading: it is actually the size of \mathbf{T} and \mathbf{L} which have decreased. Finding an optimal k value is discussed later; see “Estimating the Number of Factors in Unvalidated Models” on page 5-21.

Visualization with PCA

The columns of \mathbf{L} are the principal components, the new factors which are linear combinations of the original variables; they are also the eigenvectors of $\mathbf{X}^T\mathbf{X}$. The first loading, the m elements of the first column of \mathbf{L} , indicates how much each original variable contributes to the first principal component, PC1. The scores matrix \mathbf{T} is the projection of the samples onto the axes defined by the eigenvectors. Each sample has a coordinate on each new axis; the columns of \mathbf{T} contain these coordinates.

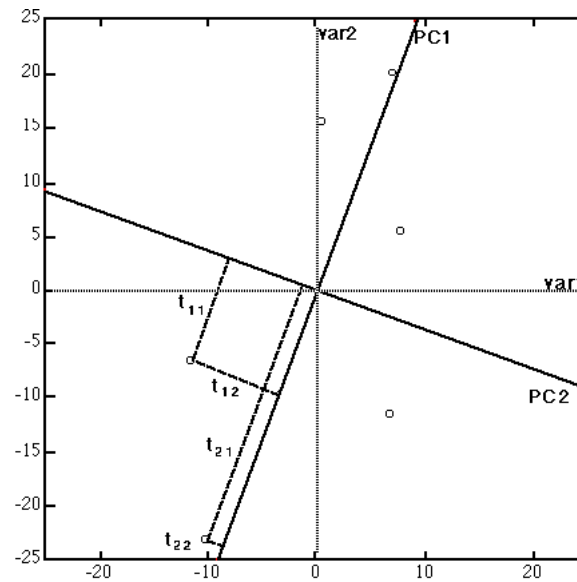
The following figures illustrate these concepts for a data set containing six samples and two variables. In the first plot, the two loadings (PC1 and PC2) are overlaid on the original variable axes. Note how PC1 aligns with the major spread in the data.

Figure 5.19
Two PC axes with labeled points (open circles)



In the next plot, the scores (t_{ij}) for two samples are marked to show how the location of data points in the original variable space can be expressed in terms of the two new axes, PC1 and PC2. While smp1 has approximately equal scores on PC1 and PC2, smp2 has a much larger score on PC1 than on PC2.

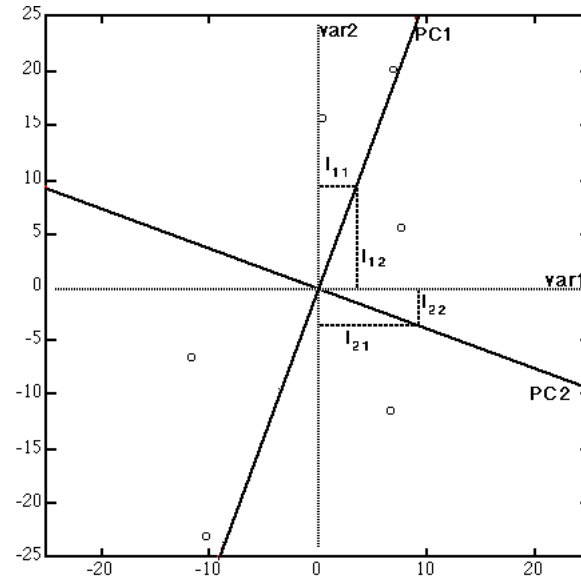
Figure 5.20
Scores: t_{ik} = score of i^{th} sample on k^{th} PC



Each loading represents the relative contribution of every original variable to a factor axis. Rarely is a factor composed of a single independent variable. Rather, each factor is a linear combination of the original independent variables. The contribution of a given variable to a given factor is shown by the projection of an arbitrary point on that factor axis onto that original variable axis. From [Figure 5.21](#), it is clear that var2 contributes

more to PC1 than does var1, which is another way of saying that var2 is more closely aligned with PC1 than is var1. A variable's loading must be between 1 and -1, and its magnitude is a measure of how much the variable *loads* into the factor. A loading of 1 implies that the variable coincides with a principal component.

Figure 5.21
Loadings: l_{kj} =
loading of the j^{th}
variable on the k^{th}
PC



The first PC is the line which minimizes the sum of squares of the distance of each sample point to the line. Thus, the first factor's direction aligns with the largest spread in the data. The second factor axis must be perpendicular to the first, by definition. Its direction is dictated by finding another line which describes the remaining variation in the data.

Modeling with PCA

Associated with each factor axis is an eigenvalue which expresses the magnitude of the variance captured by that factor. Eigenvalues decrease in magnitude with each subsequent factor because smaller and smaller amounts of the total variance remain undescribed. It is the concentration of variance in the first several PCs which permits the omission of later factors without significant loss of information. As implied by [equation 5.15](#), the transformed and preprocessed independent variable block of the data matrix is imperfectly reconstructed from the trimmed scores and loadings, that is, with some residual error, \mathbf{E} , referred to as the X Residuals in this discussion

$$\mathbf{E}_k = \mathbf{X} - \mathbf{T}_k \mathbf{L}_k^T \quad [5.16]$$

The subscripts in the above equation indicate not only that the scores and loadings matrices have been trimmed to include the first k factors but also that the residual matrix depends on k . While a proper estimate of k implies the number of phenomena giving rise to the patterns of variation, an equally important benefit is the creation of a predictive PCA model. The trimmed loadings matrix \mathbf{L}_k can be stored and used later to examine new data sets.

Suppose you have been acquiring a certain raw material from a reliable supplier and that you collect and archive spectra of the batches as delivered. You then change suppliers. By creating a PCA model from spectra of different batches of raw material from the old

supplier and then collecting a spectrum of the new material, you can compare the new material to the old. A score vector for each new sample spectrum \mathbf{x}_{new} is predicted from:

$$\mathbf{t}_{new} = \mathbf{x}_{new}\mathbf{L}_k \quad [5.17]$$

This is often described as projecting the new data into the space defined by the PCA model. If the predicted scores lie close to the scores of known-to-be-acceptable samples, the spectra do not differ significantly. The specifics are described in “Predicting in PCA” on page 5-29.

On the other hand, if a sample’s score is separated from the other samples, it may be an outlier. The question may then be asked: in what way is this sample different? Score and error contributions indicate which variables may cause the sample to be different. See “Contributions” on page 5-25 for details.

Model Validation

PCA modeling (indeed, most modeling) is based on several key assumptions.

- First, the raw material property being tracked or modeled must be manifested in the x-block data. If changes in this property do not produce changes in the data collected (spectra in the case described above), then the resulting x block variance patterns are irrelevant.
- Second, the training set (*i.e.*, the samples used to create the PCA model) should reflect normal batch-to-batch variations. Even the most reliable source cannot supply exactly the same material every time. This batch-to-batch variation is crucial for determining significance of differences.
- Third, as is always the case with statistical tests, the user must ultimately set the level of significance.
- Fourth, the PCA model must capture only the relevant variation in the training set. If factors representing random instrument variation (*i.e.*, noise) are included, the model may make misleading predictions when applied to new data. Because the noise “structure” in any new data will not be the same as in the model, this portion of the new data cannot be properly fit by the model.

This last assumption brings us back to the issue of properly estimating k , the optimal number of factors, and begs an important question: how can we distinguish a good model from a bad one?

In an ideal world, a large number of samples is available and the correctness of a PCA model is determined straightforwardly. Half of the samples are randomly placed in the training set and half are designated as the validation set. A PCA model based on a specified k value is created from the training set. Score predictions are then made on the validation set and, if they are not significantly different from the training set, the model is said to be validated and the value chosen for k is deemed appropriate. If, however, the predicted values *are* significantly different, we go back to the training set, change k and re-check the predictions until acceptable agreement is reached.

When the training and validation subsets are both representative of the population under investigation, this ideal approach guarantees model quality because the model is built from and then applied to different subsets of the same population. We can thus avoid including factors which capture random variation. On the other hand, if the validation subset has a different noise pattern than the training set, predictions will be poor if training set noise factors are included in the model.

Now back to the real world, where training sets are rarely so large as to permit a comprehensive external validation, the process just described. A compromise is cross-valida-

tion, which has a long history in the statistical community. It is computationally intensive but infinitely superior to a popular alternative, no validation at all.

Cross validation in PCA

Briefly, **leave-one-out** PCA cross validation proceeds as follows. The first sample is removed from (*i.e.*, left out of) the n member training set. For every k setting a model is constructed based on the $n - 1$ remaining (*i.e.*, left-in) samples, and a prediction is made on the left-out sample. Prediction in PCA consists of using the loadings computed from the left-out samples to generate the reconstructed X for the left-in sample. The X Residuals vector (see [equation 5.16](#)) for the left-in sample for a particular k setting is converted to a scalar by summing the square of each element of the vector. This residual sum of squares is stored for each k setting. The first sample is returned to and a different sample is removed from the training set, a new model is constructed, and a prediction is made on the second sample. This process continues until every sample has been left out once, at which point a validated residual sum of squares for each sample at each k setting is available. The matrix of validated error sum of squares for each sample at each k setting is then reduced to a vector of k validated model residual variances by averaging the n values. It is a better estimate of the true model residual variance than what would be computed without cross-validation. How this quantity is computed is shown in [equation 5.26](#). For PCA the only residue of cross-validation is this improved estimate of the model residual variance.

Note: See the [Regression Methods](#) chapter for details on PCR, PLS and CLS cross-validation.

At least three types of leave-out approaches are possible: Cross, Step, and Active Class. The first two choices are appropriate for PCA, PLS-DA, PLS, PCR and CLS models while the last applies only to regression algorithms.

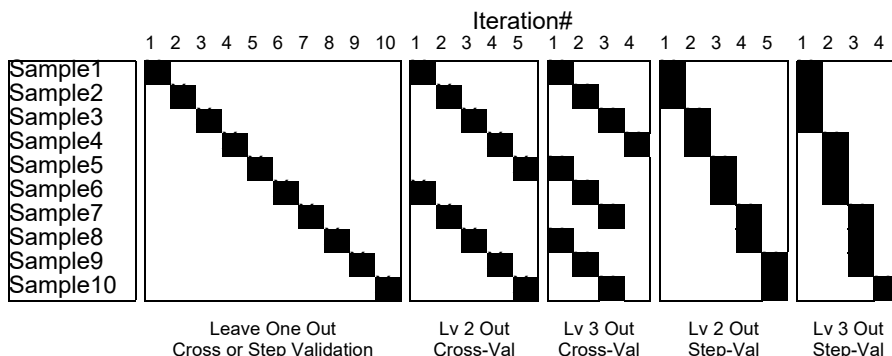
The validation described above left out one sample at a time. Because execution times may become prohibitively long for data sets containing many samples, you may choose to leave out more than one at a time. You can leave out up to half of your samples although this is not normally recommended. When more than one sample is left out at a time, the leave-out pattern must be specified. [Table 5.2](#) illustrates two different leave-out patterns for internal validation. In Pirouette, the so-called Step method leaves out sets of contiguous samples, whereas the so-called Cross method skips samples in assembling a set. In both cases, every sample is left out exactly once and, when the number left out divides unevenly into the number of samples, the final set contains fewer samples. The different leave-out patterns attempt to minimize bias caused by sample sequence:

- If replicate samples are contiguous, choose step validation
- If your sampling sequence has a periodicity such that replicates occur after predictable intervals, choose cross validation
- If the sample order is random, either technique will be satisfactory (but, see note below)
- If you are unsure, choose to leave **one** out

Note: Leaving out 1 sample at a time when replicates exist produces an over-optimistic estimate of model quality. To avoid this overly optimistic estimate, we often use leave-out-one-seventh for cross validation.

The first and second cases above, which focus on replicates, require some explanation. If replicate samples are present in the training set, all must be left out at the same time. For n contiguous replicates, this may be accomplished via step validation with a leave-out # of at least n . For replicates spaced every n samples, a cross validation leave-out pattern may work.

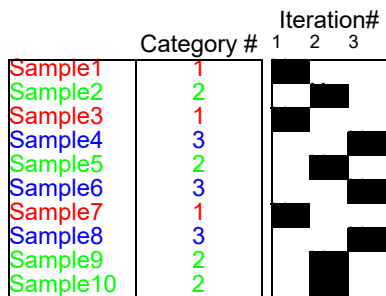
Table 5.2
Leave Out Patterns
on Step and Cross
Validation



Category validation (using the Active Class)

If your training set contains different numbers of replicates of the different samples, the leave-out patterns described above will not work; a category-based approach can be used instead (see Table 5.3). The leave-out pattern in this case is governed by a class variable activated before the algorithm run, thus the name used in Pirouette, Active Class validation. The class variable activated must have all replicates assigned to the same class. Thus, all samples belonging to one category are left out at a time.

Table 5.3
Leave Out Patterns
on Active Class
Validation



Estimating the Number of Factors in Unvalidated Models

When to stop including factors is a critical decision which crops up in all factor-based methods, *e.g.*, PCA, PCR, PLS and SIMCA. It has been the subject of much theoretical research and practical investigation. Many ways have been suggested to automate the decision (*c.f.*, Malinowski⁵).

Because a large amount of variation is compressed in the early factors, there is a point beyond which the remaining variation is essentially noise. Thus, we assume we can partition the factors into two camps: one containing relevant information and one containing irrelevant information (or noise). Finding the optimal number of factors is a matter of establishing a border between the two camps since the maximum number of factors, g , is much larger than the optimal number of factors, k . Practically this requires calculating some criterion which varies with the number of factors extracted. The hope is that when the last relevant factor is extracted, the criterion will indicate this condition, perhaps by reaching a minimum. Once the optimal number is determined, examination of (and even calculation of) additional factors can stop. Because an eigenvalue represents the magni-

tude of variation captured by the corresponding factor, it is common to define stopping criteria in terms of eigenvalues. Of the several described by Malinowski, Pirouette applies two—the IND function and the F test on reduced eigenvalues—to estimate the optimal number of factors for an unvalidated model.

The IND Function

The Indicator function, IND, is computed from what Malinowski calls the real error, RE. After k factors are extracted, the variation associated with the remaining $g-k$ factors is found by summing those $g-k$ eigenvalues. The RE is then computed:

$$RE_k = \left(\frac{\sum_{j=k+1}^g \lambda_j}{\max(n, m)(\min(n, m) - k)} \right)^{1/2} \quad [5.18]$$

From RE we can calculate IND:

$$IND_k = \frac{RE_k}{(\min(n, m) - k)^2} \quad [5.19]$$

If k is varied from 1 to g and IND is computed at each value of k , the optimal number of factors corresponds to the IND minimum.

F test on Reduced Eigenvalues

Malinowski suggests that reduced eigenvalues associated with noise are statistically equal. The k th reduced eigenvalue is defined as:

$$REV_k = \frac{\lambda_k}{(n - k + 1)(m - k + 1)} \quad [5.20]$$

Reduced eigenvalues are treated as variances. Thus, as each factor is extracted, an F test is used to decide if that reduced eigenvalue is statistically different from the remaining reduced eigenvalues. Malinowski suggests pooling the remaining reduced eigenvalues, presumably to yield an improved variance estimate:

$$REV_{pooled} = \frac{\sum_{j=k+1}^g \lambda_j}{g \sum_{j=k+1}^g (n - j + 1)(m - j + 1)} \quad [5.21]$$

An F ratio is computed:

$$F = \frac{REV_k}{REV_{pooled}} \quad [5.22]$$

and compared against values in an F table, with 1 and $g - k$ degrees of freedom at a probability level of 95% (set internally in Pirouette).

Note: *The symbol g represents the lesser of n and m , the number of samples and independent variables, respectively. If X is mean-centered or autoscaled **and** the number of samples is less than or equal to the number of independent variables, g is reduced by 1.*

Composite Eigenvalue Test

For some data sets, the IND function does not give a true minimum, instead it rises continuously. For other data sets, the F test suggests much larger optimal values than appear reasonable from visual inspection of the eigenvalues. As a result, Pirouette's estimate of the optimal number of factors is an average of the values from the two functions.

The NUMFACT Algorithm

The NUMFACT algorithm⁶ appears to be a reliable estimator of the number of factors, even for data containing highly correlated variables. This method uses a bootstrap (re-sampling) approach to determine which factors are associated with signal and which with noise.

First, the original data are decomposed by the SVD algorithm (see [page 7-4](#)). Next, the samples (rows) are resampled (with replacement) to form a second data set of the same size as the first. This set is also decomposed via SVD. Then, the eigenvectors of the bootstrapped data are projected into the factor space of the original data. The concept of this approach is that the projections of relevant factors will be large while projections of noise factors will be random and, therefore, small. A signal-to-noise computation is performed on the sum of squares of the projections for each eigenvector. Those eigenvectors with a s/n ratio above a predetermined threshold are considered relevant factors.

Estimating the Number of Factors in Validated Models

A completely different approach to estimating model size involves validation, the process of evaluating a model's predictive ability. In this case model size is inferred from stopping criteria based on predictive ability.

Optimizing Models and Finding Outliers

To date, no fail-safe method has been found for determining the optimal number of factors. You should always look carefully at several diagnostic measures at different k settings before deciding how many factors to retain. One such measure, the residual matrix E_k , was defined in [equation 5.16](#). Moreover, detection of outliers is key to estimating the optimal number of factors; these unusual samples, which have unique variance patterns, can sometimes account for (or grab) a factor. The model seeks to describe the variance patterns of typical samples, not atypical ones. Therefore, outliers must be excluded *before* successful PCA models can be built. The three quantities described below can often reveal unusual samples.

Sample Residual

A sample's residual variance follows directly from the residual matrix E_k . To make the notation less cumbersome, the subscript k will be dropped and hatted symbols will indicate a k factor approximation. The i^{th} row of \hat{E} , a vector \hat{e}_i , is the difference between that sample's original data and its k factor estimate, \hat{x}_i :

$$\hat{\mathbf{e}}_i = \mathbf{x}_i - \hat{\mathbf{x}}_i \quad [5.23]$$

Note: For PCA, even when cross-validation is performed, the validated X Residuals are not used to compute the Sample Residual. They are used only to compute the model residual (variance). Thus, it is possible to compute the Sample Residuals shown in the Outlier Diagnostics object from the data in the X Residuals object by using [equation 5.24](#). This is not true for PLS, PCR or CLS.

A sample's residual variance is then:

$$\hat{s}_i^2 = \frac{\hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T}{m - k} \quad [5.24]$$

In Pirouette, the square root of sample residual variance is called the *sample residual*:

$$\hat{s}_i = \left(\frac{\hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T}{m - k} \right)^{1/2} \quad [5.25]$$

The model residual variance is calculated for the whole training set:

$$s_0^2 = \frac{1}{n} \sum_i^n \hat{s}_i^2 \quad [5.26]$$

Other terms for s_0^2 include *model error sum of squares* and *model residual*. The *Q-statistic* is a related diagnostic on the sum of squared errors (see “[Q Statistic](#)”, below).

If a particular sample residual variance is larger than the model residual variance, it is natural to wonder if the sample is an outlier, *i.e.*, it might not belong to the same population as the other samples in the training set. An F test is used to decide if two variances differ significantly, the appropriate ratio being:

$$F_i = \frac{\hat{s}_i^2}{s_0^2} \quad [5.27]$$

If the left-hand side of [equation 5.27](#) is set equal to a critical value extracted from an F table (based on 1 and $n - k$ degrees of freedom and a user-specified probability), a critical sample residual can be determined by rearrangement:

$$s_{crit} = s_0 (F_{crit})^{1/2} \quad [5.28]$$

This then becomes a threshold for deciding whether a sample residual is “too large”. If a sample residual exceeds s_{crit} , that sample may be an outlier.

Q Statistic

Sometimes referred to as the Squared Prediction Error (SPE), Q is related to the sample residual variance ([equation 5.24](#)), but without normalization.

$$Q = \hat{\mathbf{e}}_i \hat{\mathbf{e}}_i^T \quad [5.29]$$

It is used in the same way as the sample residual to evaluate whether a sample might be an outlier and is frequently tracked in statistical process monitoring settings. The critical value of Q can be determined from its approximate distribution⁷.

Note: *Computing the critical value of Q can be time-consuming. Thus, obtaining the threshold is optional, and is controlled in the Run Configure dialog box (e.g., see Figure 16.24, on page 16-22).*

Probability

Another way to flag unusual samples is by determining the probability associated with the quantity in equation 5.27 assuming an F distribution with 1 and n - k degrees of freedom. As a sample's probability approaches 1, the chance it is an outlier increases.

Consider, for example, a probability cutoff of 95%. Our null hypothesis is that the two variances in the equation are equal. Thus, if the probability value corresponding to the F value exceeds 95%, then the hypothesis is rejected, and we infer that the sample was not drawn from the same population.

Mahalanobis Distance

For each sample, an overall measure of variability is computed: the Hotelling's T² statistic, a multivariate version of the t-test for the equivalence of means⁸. Although T² can be computed directly from **X**, in multivariate analysis it is usually based on the scores. The distance from the multivariate mean, commonly referred to as the Mahalanobis distance, is computed from the k factor score vector:

$$MD_i = (\mathbf{t}_i - \bar{\mathbf{t}})\mathbf{S}_k^{-1}(\mathbf{t}_i - \bar{\mathbf{t}})^T \quad [5.30]$$

where **S** is the scores covariance matrix and $\bar{\mathbf{t}}$ is the mean score vector. Assuming that Mahalanobis distance is normally distributed, a critical value MD_{crit} can be determined from an F distribution.

$$MD_{crit} = \frac{k(n-1)}{n-k} F_{\alpha, k, n-k} \quad [5.31]$$

If a sample's Mahalanobis distance exceeds MD_{crit}, that sample may be an outlier.

Note: *The critical value for the MD has changed slightly from earlier versions of Pirouette to conform with work in the process field. Previously it was based on a Chi squared distribution.*

Contributions

Pirouette's outlier diagnostic object contains several measures of how well a sample fits a model. Some of these measures relate to how close a sample's score is to the center of the model's score space. Others relate a sample's X Residual to the model's average X Residual. However, regardless of whether the diagnostic is of type "in-model" (i.e., scores based) or "out-of-model" (i.e., residuals based), it is natural to wonder **which** variables **contribute** to the sample's unusual outlier diagnostic value. This information could be helpful in determining which of the original measurements are causing the problem, particularly in discrete variable control scenarios, where sensors can fail/degrade. Below

the two flavors, Score Contributions and Error Contributions, are described. This concept of contributions comes from the quality control field.

A sample's score on factor k can be written as a sum of m terms, one for each variable j .

$$t_k = \sum_{j=1}^m x_j L_{jk} \quad [5.32]$$

The *score contribution*³, ct , retains each of these terms in a vector of m values; there is one vector for each sample at a given factor setting:

$$ct_k = x_j L_{jk} \quad [5.33]$$

for $j=1, \dots, m$. For an example, see “Contributions” on page5-40.

Large score contributions may indicate variables that produce that sample's unusual score, explaining how it differs from the model.

In some situations, a sample may appear unusual in more than one score, thus an overall or total contribution CT can also be computed⁴ by summing the scaled contribution on each score:

$$CT_j = \sum_{a=1}^k \frac{t_a}{\lambda_a} ct_{a,j} \quad [5.34]$$

The scale factor is the score value divided by the variance in that factor.

Samples having a variance structure different from the model will have large X residuals (see equation 5.23); the Q statistic (see “Q Statistic” on page5-24) flags samples with unusual residuals. We can compose an *Error Contribution* vector, ce , as the components to Q.

$$ce_j = (x_j - \hat{x}_j)^2 \quad [5.35]$$

Thus, if a sample is unusual on an out-of-model measure (e.g., sample residual or Q), the contributing variables may be determined from the error contributions.

To summarize, a sample's score on a particular factor can be broken down into contributions from each variable, producing Score Contributions. When f factors are considered, a sample's Hotelling's T^2 can be broken down into contributions from each variable, producing the Total Contributions object. A sample's residual sum of squares indicates the contribution of each variable, producing the Error Contributions object.

Error Sum of Squares

The sample residual variance described by equation 5.24 can be assembled into a column vector with an element for each sample. If each element is squared and the sum corrected for the degrees of freedom, a predicted residual error sum of squares (i.e., PRESS) is calculated for the x block in the training set:

$$PRESS = \sum_i^n \hat{s}_i^2 \quad [5.36]$$

As the number of factors increases, the quality of the approximation must increase and PRESS approaches zero monotonically.

Validated Error Sum of Squares

If the sample residual variance described by [equation 5.24](#) arises during cross-validation, a corresponding validated predicted residual error sum of squares (*i.e.*, VPRESS) is found:

$$VPRESS = \sum_i^n \hat{s}_{icv}^2 \quad [5.37]$$

The cv subscript is added to minimize confusion with the quantity defined in [equation 5.36](#). As the number of factors increases, VPRESS does not always decrease monotonically. It may reach a minimum and increase, indicating the models's predictive ability does not improve with the inclusion of more factors.

An F test can determine if two VPRESS values are significantly different.⁹⁻¹⁰ Note that we need only compare those models having fewer factors than the minimum PRESS model. For the typical case of more variables than samples,

$$F_k = \frac{VPRESS_k - VPRESS_{min}}{VPRESS_{min}} \frac{n}{n-k} \quad [5.38]$$

with F_k compared against tabulated values using k and (n-k-1) degrees of freedom and a probability level of 95% (set internally in Pirouette). If there is no significant difference, the more parsimonious model, *i.e.*, the one with fewer factors, is chosen. When cross-validation is performed in Pirouette, the number of optimal factors is based on such an F test. Like the eigenvalue-based estimate, it is not carved in stone—you may override the default value.

Modeling Power

Sample residual and Mahalanobis distance are both sample-oriented measures. Modeling power varies with k but is variable-oriented. Typically, it is *not* helpful in determining the optimal number of factors to retain but does point out important variables.

For this quantity we first compute a variable residual variance, \hat{s}_j^2 , using the j^{th} column of the residual matrix \mathbf{E} :

$$\hat{s}_j^2 = \frac{\hat{\mathbf{e}}_j^T \hat{\mathbf{e}}_j}{n-k-1} \quad [5.39]$$

The total variance of that variable is:

$$s_{0j}^2 = \frac{1}{n-1} \sum_i^n (x_{ij} - \bar{x}_j)^2 \quad [5.40]$$

Modeling power is defined as:

$$MP_j = 1 - \frac{\hat{s}_j}{s_{0j}} \quad [5.41]$$

As the power of a variable to model information in the data increases, MP approaches 1; as it decreases, MP approaches 0 (it can also sometimes become negative). Even with random data, some variables will exhibit high modeling power so an absolute threshold cannot be specified. Instead, the different variables should be compared based on their relative modeling power.

NIPALS

There has been considerable research into efficient algorithms for the computation of principal components. We have chosen NIPALS (Nonlinear Iterative Partial Least Squares¹¹⁻¹⁴) for PCA because it finds the first k principal components without computing all factors. (In fact, it is very inefficient at computing *all* factors.) For details on NIPALS and related approaches, see the “Reading List” on page 5-47.

Varimax Rotation

Once a decision has been made about the optimal number of factors in a data set, a natural question arises about the meaning of these so-called abstract factors/latent variables. This leads to an area of study called factor analysis, which is often incorrectly assumed to be synonymous with PCA. Factor analysis has a rich history, a vast literature, and a peculiar nomenclature. Typically, an application of factor analysis starts with PCA but does not end there. The next step is often a rotation of factors. Pirouette implements several types of post-PCA orthogonal rotations. Rotations are an attempt to make factors less abstract by aligning them with original variables; see “Rotated Loadings, Scores and Eigenvalues” on page 5-41 for more discussion.

Varimax rotation maximizes the variance of the loadings by sequentially rotating PCA loadings pair-wise. The rotated loadings matrix **F** is calculated from:

$$\mathbf{F} = \mathbf{R}\mathbf{L}^* \quad [5.42]$$

where **R** is a rotation matrix and **L*** is the PCA loadings matrix or a modification of it described below. Various methods have been proposed for finding and updating the rotation matrix, **R**. We use the approach given in Harmon¹⁵. Once a trial value for **R** has been found, the matrix **F** is computed and a pseudo-variance of **F**, known as the simplicity, is determined:

$$simplicity = m \sum_i^k \sum_j^m f_{ij}^4 - \sum_i^k \left(\sum_j^m f_{ij}^2 \right)^2 \quad [5.43]$$

where the summations are over the k principal components and the m original variables. The rotation matrix is then iteratively modified until the simplicity no longer increases.

Pirouette’s several Varimax algorithms differ only in the manner in which the loadings are normalized and/or weighted before and after rotation¹⁶. The normalization is accomplished by dividing each loading vector element by h_j , the square root of the communality of the j^{th} variable. Communality is a measure of factor variance for a given variable:

$$h_j^2 = \sum_i^k l_{ji}^2 \tag{5.44}$$

The weighting is applied by multiplying each loading by its singular value, s_i . A singular value is the square root of the corresponding eigenvalue.

After rotation, the normalization is removed by multiplying by the communality. The weighting is reversed by dividing by the square root of the communality of each factor, $h_{f(i)}$, which is the singular value of the newly-rotated factor:

$$h_{f(i)}^2 = \sum_j^m l_{ji}^2 \tag{5.45}$$

The loadings treatments described above are defined in Table 5.4. The last method, Weighted-Normal, is implemented as described in the mainframe program ARTHUR¹⁷.

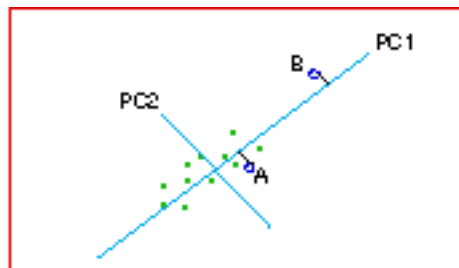
Table 5.4
Treatments of loadings during Varimax rotation

Varimax Method	Before Rotation	After Rotation
Raw	none	none
Normal	$L^* = \frac{L}{h}$	$L^R = Fh$
Weighted	$L^* = (SL^T)^T$	$L^R = \frac{F}{h_f}$
Weighted-Normal	$L^* = \frac{(SL^T)^T}{h}$	$L^R = \frac{Fh}{h_f}$

Predicting in PCA

Predictions are made in PCA by projecting new samples, the unknowns, into the PC space defined by the k training set loadings, where k is the number of optimal factors. The purpose of prediction in PCA is to decide if the new sample differs significantly from the training set. This decision is based mainly on the magnitude of the X residuals when the new sample is projected into the model factor space: samples significantly different will have large residuals. However, to address the scenario shown in the next figure, the matter becomes more complicated. There, two unknowns A and B have been projected into the space defined by the first two principal components of the training set. Review the discussion of “Sample Residual” on page 5-23 before considering the remarks below.

Figure 5.22
A two PC model with unknowns A and B



Unknown A is clearly in the region of the training set samples (the small green dots) while B is not. Yet suppose that both have sample residuals (calculated from the square root of [equation 5.24](#)) which are less than s_{crit} .

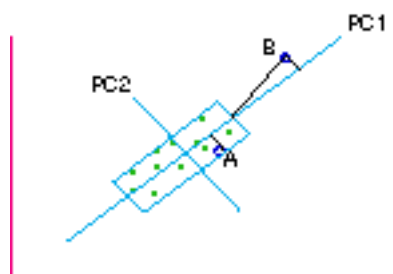
Note: Any statement about the magnitude of residuals cannot be verified by examining [Figure 5.22](#). Residuals are computed in the multidimensional data space but [Figure 5.22](#) contains points plotted in two dimensional PC space.

In an attempt to flag B as more unusual than A, the sample residual computation can be modified during prediction based on the scores hyperbox, described in the next section.

Score Hyperboxes

It is convenient to construct a virtual box around the scores on each PC based on the training set values. Because the edges of this hyperbox are the minimum and maximum score value in each factor dimension, the hyperbox is seldom symmetrical about the factor space origin. The following graphic shows such a hyperbox.

Figure 5.23
Figure 5.22 with
hyperbox



These bounds are roughly comparable to the confidence ellipse shown in 2D score scatter plots (see [Figure 5.31](#), on page 5-35). When a sample projected into the model space falls beyond the edge of the hyperbox, this distance can be used to augment the effective sample residual (see below).

Augmenting the Sample Residual in Prediction

Point B clearly resides outside the hyperbox in [Figure 5.23](#). Increasing its sample residual by an amount proportional to its distance to the hyperbox edge increases the chance that B will be flagged as an outlier. An augmented sample residual (also called a distance) is calculated from:

$$s_u' = \left(s_u^2 + \sum \frac{s_0^2}{s_t^2} (t_u - t_{lim})^2 \right)^{1/2} \quad [5.46]$$

The summation is over the k included principal components and the ratio inside the summation symbol makes the units match.

Because the augmented sample residual is also involved in the calculation, the probability of unknowns outside the hyperbox also increases due to the ratio:

$$F_u = \left(\frac{s_u'^2}{s_0^2} \right) \quad [5.47]$$

The decision to augment the sample residual is controlled in Prediction Preferences (see “Prediction” on page 10-19). By default, it is turned off. Before changing the default setting, a prudent user should contemplate and understand the ad hoc nature of this augmentation.

Mahalanobis Distance in Prediction

For each unknown, a Mahalanobis distance can be computed from the its k factor score, t_{ui} :

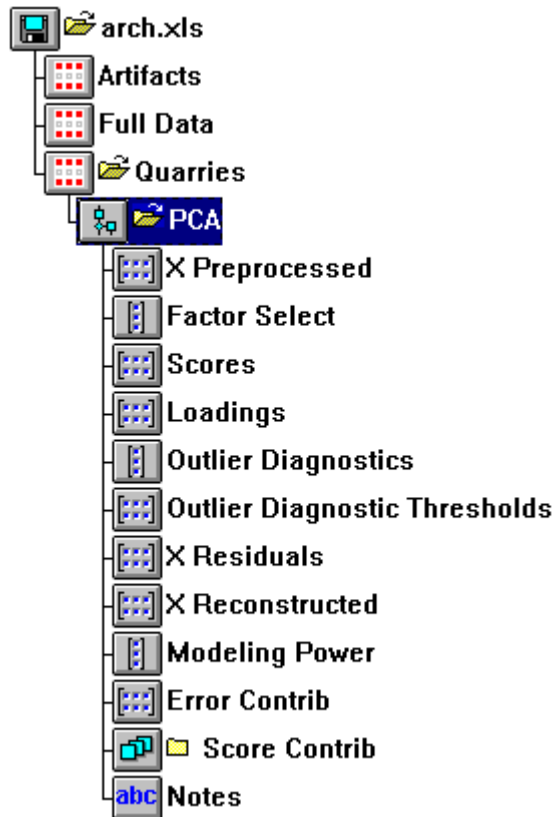
$$MD_{ui} = (t_{ui} - \bar{t})S_k^{-1}(t_{ui} - \bar{t})^T \quad [5.48]$$

where S_k is the covariance matrix of the training set scores trimmed to k factors, and \bar{t} is the mean training set score vector.

RUNNING PCA

The options associated with PCA are described in “PCA Options” on page 16-22. When the algorithm runs, it computes and displays many entities (see Figure 5.24) which can help you explore the relationships between samples, find sample outliers, choose the optimal number of factors and make decisions about excluding variables. Each is described below along with ideas about how to examine them. If you are most interested in the visualization aspects of PCA, your task is simple: focus on the Eigenvalues portion of the Factor Select object, the Scores, and the Loadings.

Figure 5.24
PCA computed
objects



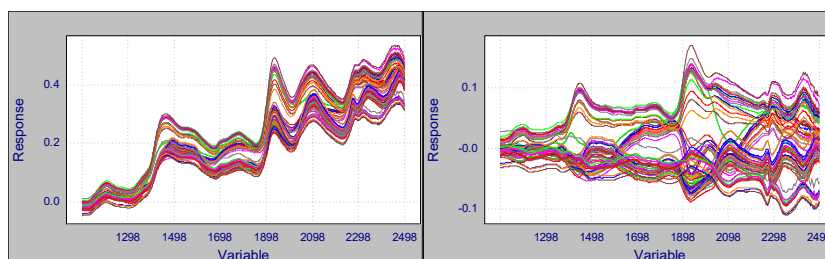
In addition to the computed objects, information necessary to make predictions is stored as a model. A model can be used as soon as it has been created or it can be stored in a file separate from the training set data and reloaded later to make predictions on future samples (see [“Saving Models” on page 15-6](#)). A Pirouette PCA model is more than just a loadings matrix trimmed to k factors. It also contains information about which variables were excluded and what transforms/preprocessing options were chosen so that future samples are treated in the same way as the training set. Model building is an iterative process. You seldom run the PCA algorithm just once and immediately start making predictions. Instead you spend most of your time optimizing your model, that is, finding the “best” set of samples, variables and algorithm options.

X Preprocessed

This object contains the actual data processed by the PCA algorithm. These are the values after transforms and preprocessing have been applied. It is often useful to examine this object in conjunction with others to understand what features are important and to contrast it with the raw (that is, original) X block.

The figure below is an example that shows raw data on the left and the X Preprocessed data on the right.

Figure 5.25 Raw data and X Preprocessed profiles



Factor Select

Decisions about the appropriate number of factors in a PCA model can be based on the eigenvalues and error sums of squares as described in [“Estimating the Number of Factors in Unvalidated Models” on page 5-21](#) and [“Estimating the Number of Factors in Validated Models” on page 5-23](#) respectively. Accordingly, the Factor Select object contains both types of computations for every factor extracted. Its table view is shown in [Figure 5.26](#).

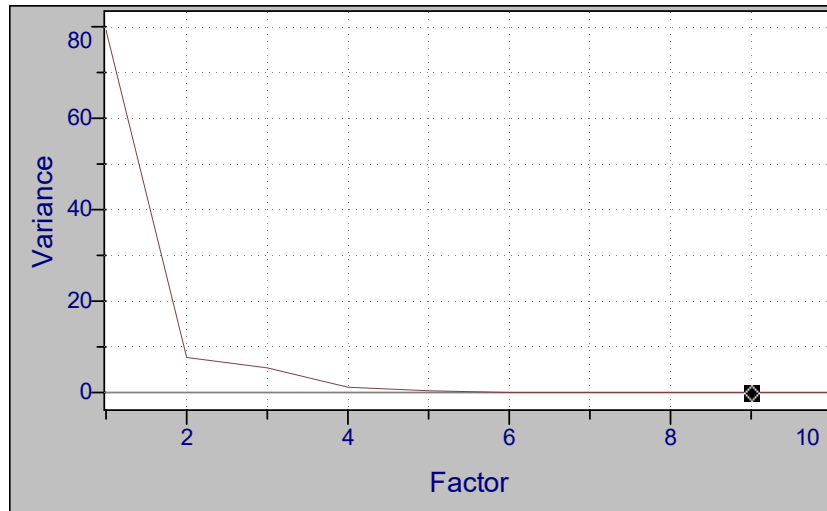
The first three columns are related to the data set eigenvalues. The Variance column holds the eigenvalue associated with each PC, the variation captured by that PC. The Percent column expresses the variation as a percentage; the Cumulative column expresses the percentage cumulatively. The largest eigenvalue and thus, the largest variance are always associated with PC1. As more factors are computed, the associated variance decreases. In this case, the first PC accounts for roughly 86% of the total variance and, by the third component, over 99% of the x block variance has been captured.

Figure 5.26
Factor select table
view

		1	2	3	4	5
		Variance	Percent	Cumulative	Press Val	Press Cal
1	Factor1	79.071442	84.832207	84.832207	14.849749	14.137779
2	Factor2	7.702389	8.263549	93.095757	7.424602	6.435390
3	Factor3	5.232176	5.613367	98.709122	1.381209	1.203214
4	Factor4	0.923141	0.990396	99.699516	0.387374	0.280074
5	Factor5	0.235133	0.252264	99.951782	0.057866	0.044940
6	Factor6	0.014573	0.015635	99.967415	0.044825	0.030367
7	Factor7	0.008969	0.009622	99.977036	0.037057	0.021398
8	Factor8	0.007835	0.008406	99.985443	0.024974	0.013563
9	Factor9	0.005131	0.005505	99.990952	0.014445	0.008433
10	Factor10	0.001261	0.001353	99.992302	0.013543	0.007172

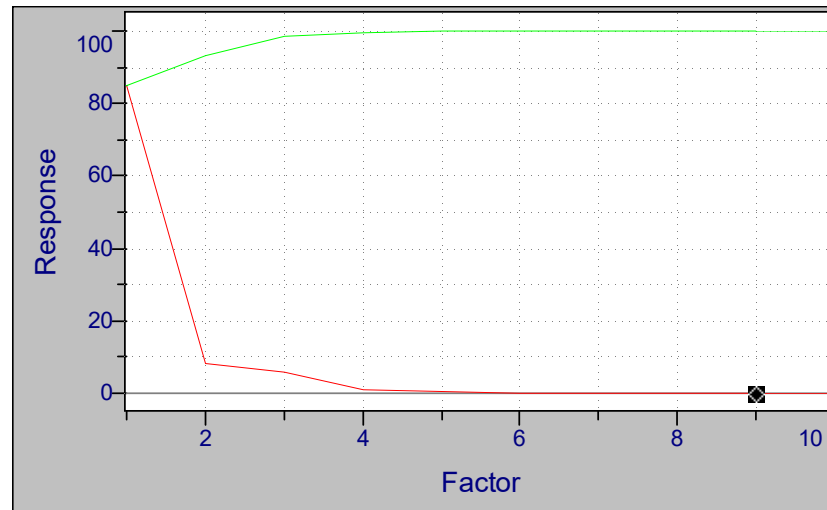
The default view of the Factor Select object, a line plot of the Variance column, appears in the figure below.

Figure 5.27
Eigenvalues vs.
Factor #



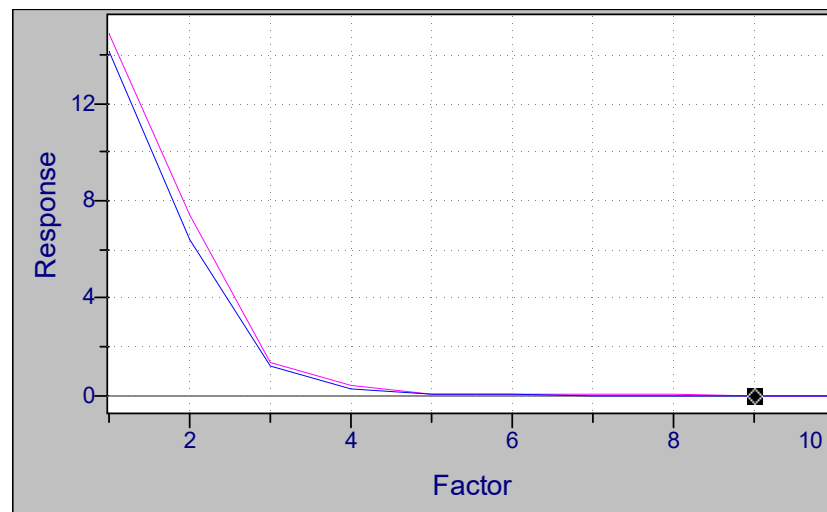
In some situations, it is more informative to plot percentages rather than raw variance. In [Figure 5.28](#) both percent and cumulative percent line plots are shown. The only difference between the Variance trace above and the Percent Variance trace below is vertical scale.

Figure 5.28
Percent Variance
and Cumulative
Percent Variance



The remainder of the Factor Select object holds error sum of squares computations. If no validation was specified, only Press Cal (defined in [equation 5.36](#)) is shown; otherwise both Press Val (defined in [equation 5.37](#)) and Press Cal appear. In the next figure, Press Cal and Press Val are shown superimposed in a line plot.

Figure 5.29
Press Cal and Press
Val

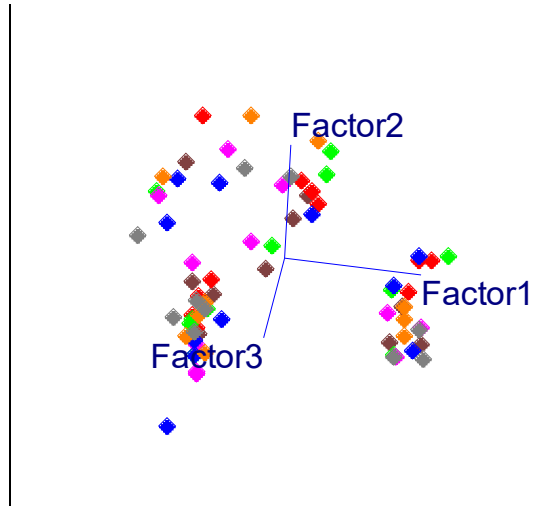


A graphical handle in the appearance of a diamond is always present in a Factor Select line plot, indicating the current setting of k , the number of factors (*i.e.*, PCs) retained. Pirouette initially suggests a value for k based on one of two stopping rules. If cross-validation was not specified in the run configuration, the diamond position is set by the process outlined in [“Estimating the Number of Factors in Unvalidated Models”](#) on page 5-21. If cross-validation *was* specified, the diamond position is determined from the Press Val as described in [“Validated Error Sum of Squares”](#) on page 5-27. Regardless of how it was positioned, the user must always critique the setting. For this reason, the diamond’s position is easily changed. The diamond jumps to wherever the mouse is clicked along the x axis. Moreover, any computed result depending on the number of retained PCs automatically updates to reflect the new setting. Typically, the number of factors is changed and the effect monitored on some or all of the objects discussed below.

Scores

Scores are integral to exploratory analysis because they show intersample relationships. A 3D view is particularly informative as it offers an interactive mode: the Spinner Tool, the arrow keys, or the Spin Control Buttons change the viewpoint continuously. A 3D scores plot is shown below.

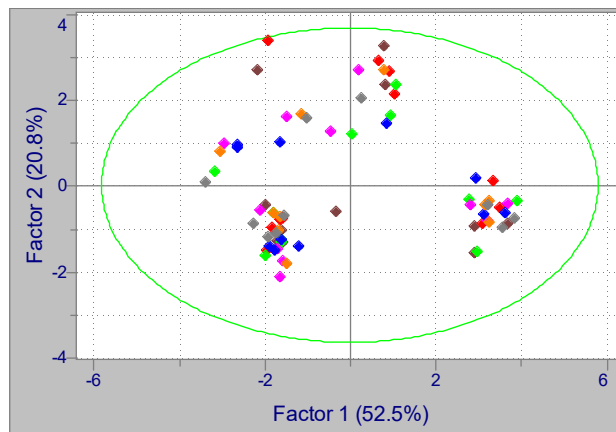
Figure 5.30
A 3D scores plot



When examining scores plots, the user must keep in mind the purpose of the investigation. If a PCA model is the goal (*i.e.*, single category classification), the scores should not cluster strongly. If the ultimate goal is multi-category classification and PCA is being used as an exploratory technique, sample groupings must correspond to known categories. If regression is the goal, a single, homogeneous swarm of samples is desirable. The existence of more than one cluster in this case suggests that more than one population is present.

When scores are shown in a 2D scatter plot, a confidence ellipse is superimposed. During PCA modeling, the ellipse represents the 95% confidence level.

Figure 5.31
2D Scores plot



These confidence boundaries are derived from the score variance. The ellipse is centered at the origin of the two score dimensions to be displayed. Axis lengths (+/-) are computed from

$$A_i = \sqrt{s_i^2 \cdot f \cdot df} \quad [5.49]$$

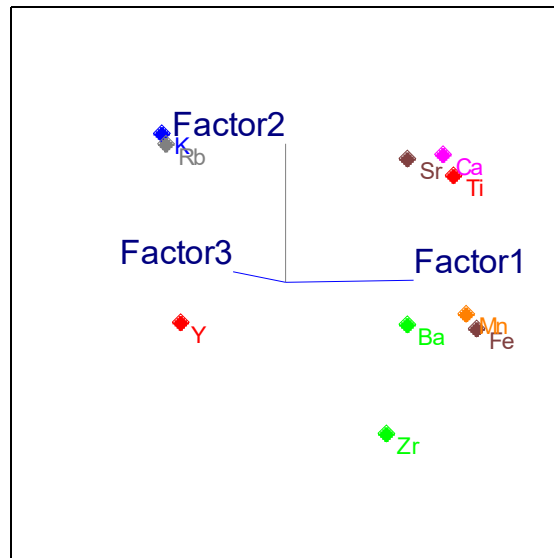
where A_i is the ellipse axis for the i th PC, s_i are the corresponding score standard deviations, f is the critical F value ($k, n-k, \alpha$) at k factors and a probability level α , and df are the degrees of freedom,

$$df = k(n^2 - 1)/(n(n - k)) \quad [5.50]$$

Loadings

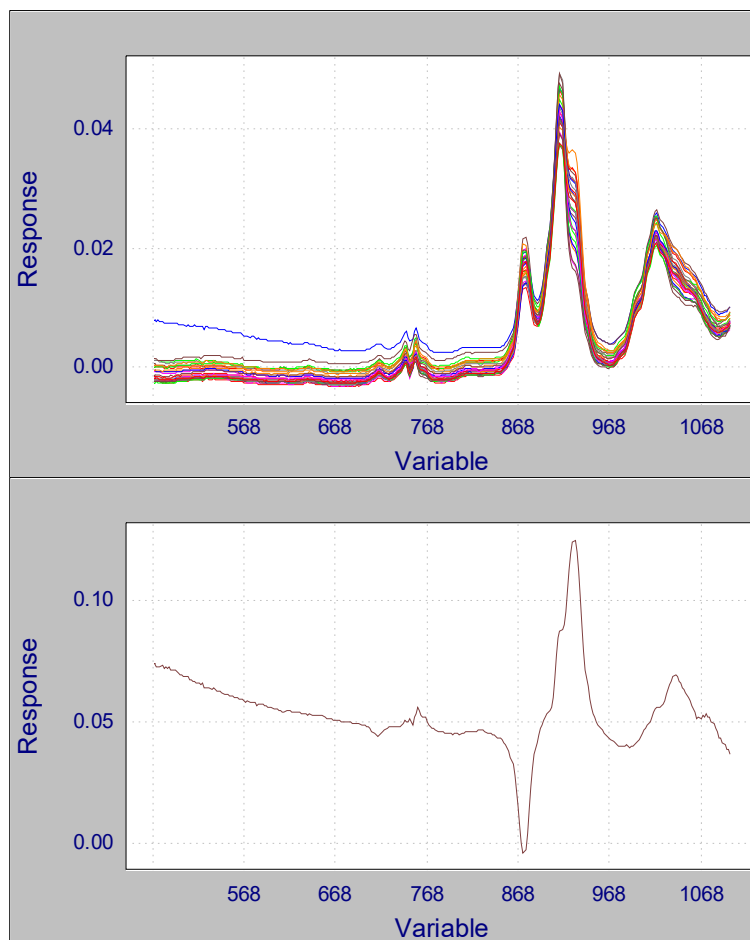
The Loadings object indicates which variables are most important and which contribute little to each PC. A 3D loadings plot is shown next.

Figure 5.32
A 3D loadings plot



Directly comparing line plots of the original data and loadings lets you see which data features are captured by a particular PC. Such a comparison is illustrated below.

Figure 5.33
Comparing line plots
of raw data and the
first loading

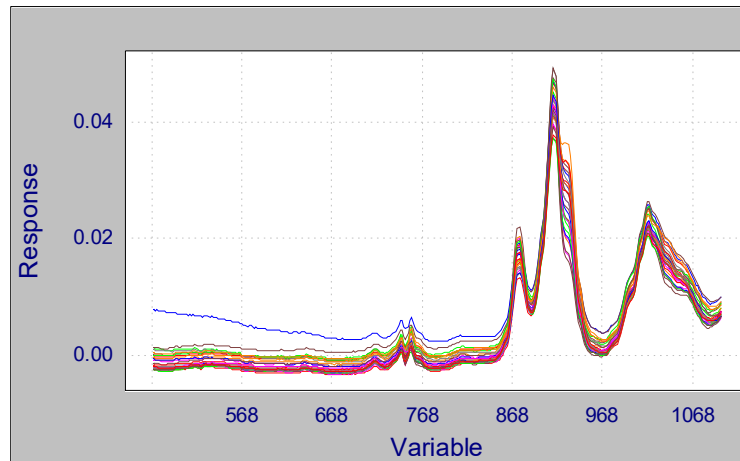


The two significant features in the plot of the first loading correspond to the peaks at 874 and the shoulder peaks at 928 nm, respectively. Because the first factor in this data set accounts for over 70% of the variance, the most important descriptive information about the data, then, is the inverse correlation between these two peaks.

X Reconstructed

An estimate of the original preprocessed independent variable block based on the first k factors can be calculated using [equation 5.15](#). If the effect of preprocessing is undone by dividing each row by the training set standard deviation vector and adding to each row the training set mean vector, the result is the X Reconstructed object, shown below in a line plot.

Figure 5.34
X Reconstructed

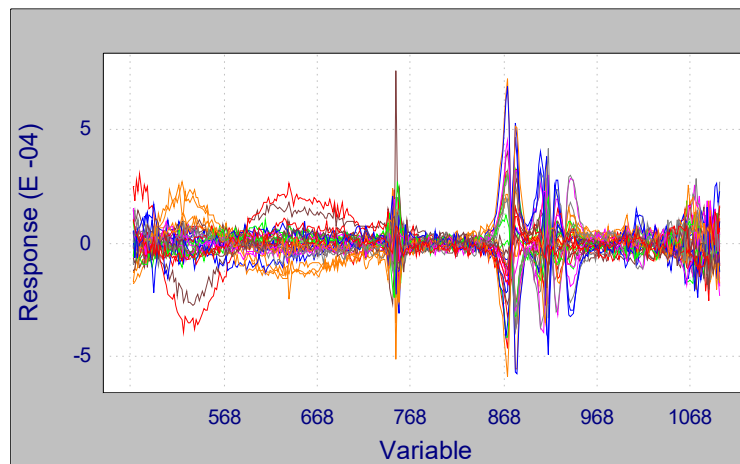


The X Reconstructed object can be compared to the unprocessed data, that is, the data *after transforms have been applied*. As more factors are added, the shape of the transformed data and the reconstructed data converge; the reconstruction becomes more accurate.

X Residuals

The X Residuals object is the difference between the transformed data and its k factor estimate, X Reconstructed. It is good for pinpointing samples/variables poorly fit by a k factor model. It can also verify that the difference between the transformed data and the reconstructed data is of reasonable magnitude compared to the uncertainty of the measurement technique. A line plot of this object is shown below.

Figure 5.35
X Residuals



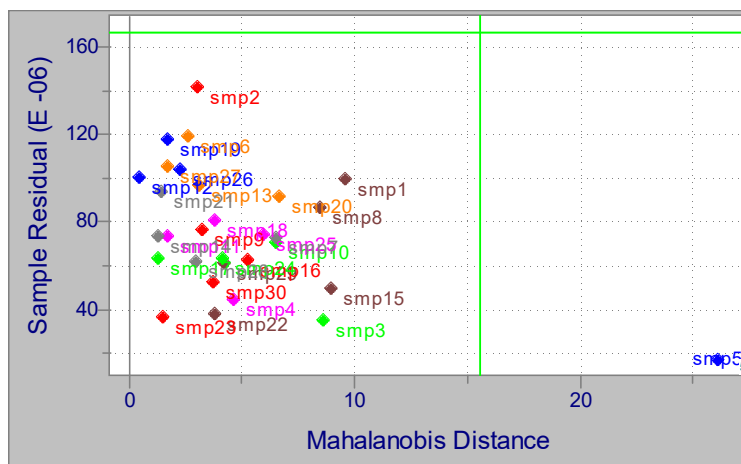
In evaluating the appropriateness of a k-factor model, look for structure in the X residuals object. Where structure distinguishable from noise occurs, the corresponding variables have not been completely modeled.

Outlier Diagnostics

Pirouette's Outlier Diagnostics object includes measures of how well a sample is approximated by k factors: Sample Residual, Mahalanobis Distance, F Ratio, Probability, and the Q statistic. The quantities are discussed in [“Sample Residual”](#) on page 5-23, [“Mahala-](#)

nobis Distance” on page 5-25, “Probability” on page 5-25, and “Q Statistic” on page 5-24, respectively. A 2D plot of the first two measures, is shown below.

Figure 5.36
Outlier Diagnostics



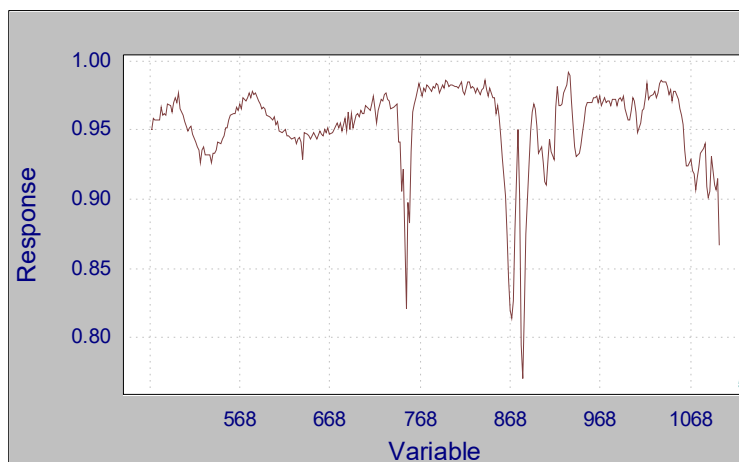
Samples falling outside one or both of the thresholds are potential outliers. Because the sample residual threshold is based on a 95% probability limit (set internally in Pirouette), 5% of *normal* samples would be expected to fall outside that cutoff. For this reason in a large data set, samples exceeding only one threshold slightly may be normal. However, samples lying either significantly beyond one threshold or beyond both are more likely to be outliers. You should examine these samples closely, try to understand how they differ from the others and consider rerunning PCA with those samples excluded.

The Probability object, which is based on the same information as the Sample Residual, allows decisions to be made using a metric which ranges from 0 to 1. The magnitude of a probability, however, is highly dependent on the degrees of freedom. Typical spectroscopic data violates the assumption of uncorrelated variables made in deriving the F distribution on which the probability computation is based. The degrees of freedom used in the calculation is consistent but almost certainly incorrect. Probabilities should be compared relatively and appropriate cutoffs developed over time as the user gains experience with the data source.

Modeling Power

A line plot of modeling power vs. variable number is shown below. Modeling power can be helpful for finding variables to consider excluding. Note, however, that because modeling power changes as a function of the number of PCs, variables poorly modeled at one setting may be adequately modeled at another.

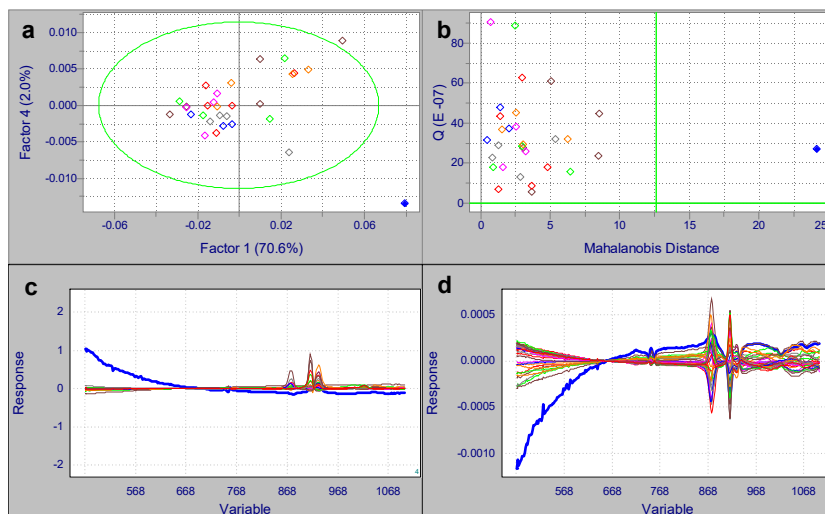
Figure 5.37
Modeling power line plot



Contributions

When a sample is suspected of being an outlier, it may be interesting to understand what makes the sample different from the other samples in the data set. Examination of the Mahalanobis distance may tell us if a sample does not fit the model well, and the score plots should indicate on which factor (or factors) the deviation from the model occur. The individual or overall score contributions (see [equation 5.32](#)) show where the sample's profile differs from the main PCA model.

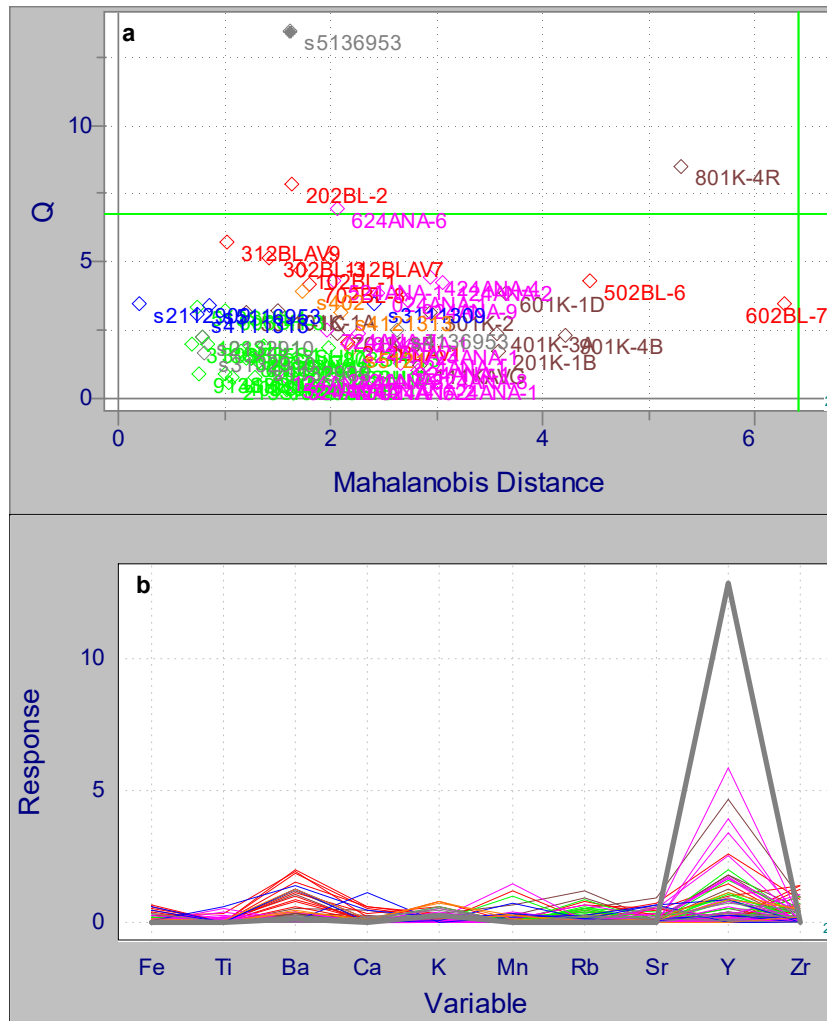
Figure 5.38
Score contributions for an outlier sample (highlighted), (a) scores plot, (b) outlier diagnostics, (c) overall score contributions, and (d) factor 4 score contributions



Here the unusual sample is clearly aberrant in the factor 4 direction, thus observation of the factor 4 contribution shows in what variable region the sample differs from those in the training set. The overall contribution shows similar information.

On the other hand, if a sample has a high sample residual or Q statistic, it is an indication that it is in a space different from the model. These two diagnostics measure how far a sample is from the model; the error contributions may indicate which variables cause this discrepancy.

Figure 5.39
Error contributions,
(a) Outlier
diagnostics
indicating potential
outlier, and (b) error
contribution plot
showing cause for
outlier (highlighted)
is in 9th variable



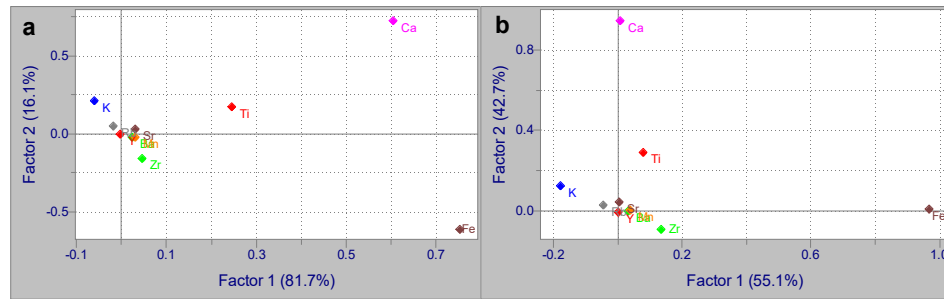
Rotated Loadings, Scores and Eigenvalues

Ultimately, the physical meaning of a factor is inferred from the original variables “loading” most heavily onto it. By considering them as a group, we may be able to understand what phenomena give rise to the loadings. One means to accomplish this is via Varimax rotation.

In Varimax, pairs of loadings are rotated in a plane orthogonal to all other factors so as to “simplify” the factor description of the variables. The simplification is based on a least-squares criterion of the loadings¹⁵. The result of this simplification is that high loading variables are accentuated and low loading variables are minimized. This has been characterized as an approach where “the rich get richer and the poor get poorer.” Ideally, rotated loadings point out variables which group together. Note how in the rotated loadings plot of Figure 5.40b, the variables, particularly the two isolated variables, tend to align with the axes more so than in the unrotated loadings. Note also that the amount of variance in the rotated loadings is much more similar than in the unrotated loadings.

5 Exploratory Analysis: Principal Component Analysis

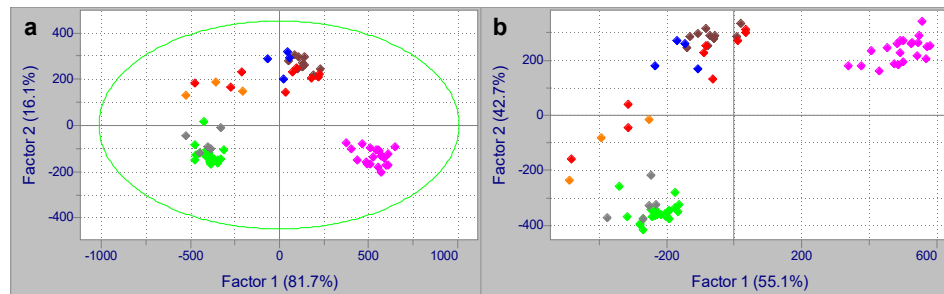
Figure 5.40
Loadings (a)
unrotated (b) rotated



Imagine creating a correlation matrix to identify highly correlated variables. Varimax acts on the correlation information to create a new set of axes that are more “pure” in terms of the original variables. By noting which variables increase in importance, you may be able to infer underlying causes of the variation.

When Varimax rotation is performed, eigenvalues, scores and loadings for the unrotated factors are first computed. The designated number of loadings are then rotated, and the corresponding rotated scores and rotated eigenvalues computed. The following figure compares the unrotated and rotated scores associated with the loadings in the previous figure.

Figure 5.41
Scores associated
with Figure 5.40 (a)
unrotated (b) rotated



The next figure shows the eigenvalues associated with the loadings and scores of the previous figures. Variance is spread more uniformly over the eigenvalues after rotation due to the rotation away from the maximum variance directions found by PCA. This is evident in the first two principal components of Figure 5.42: from 81 and 16% of the total variance, rotation causes the relationship to arrive at 55 and 42% of the total. Thus, the first two factors now “share” the bulk of the variance more than originally.

Figure 5.42
Eigenvalues
associated with
Figure 5.40
(a) unrotated
(b) rotated

		1	2	3
a		Variance	Percent	Cumulative
1	Factor1	11810908	81.678741	81.678741
2	Factor2	2334048.2	16.141191	97.819931
3	Factor3	170377.21	1.178250	98.998184
4	Factor4	67367.203	0.465880	99.464066
5	Factor5	59778.425	0.413400	99.877464
6	Factor6	6060.7500	0.041913	99.919380
7	Factor7	4812.2758	0.033279	99.952660
8	Factor8	3506.0935	0.024247	99.976906
9	Factor9	2048.9936	0.014170	99.991074
10	Factor10	1289.2429	0.008916	99.99992

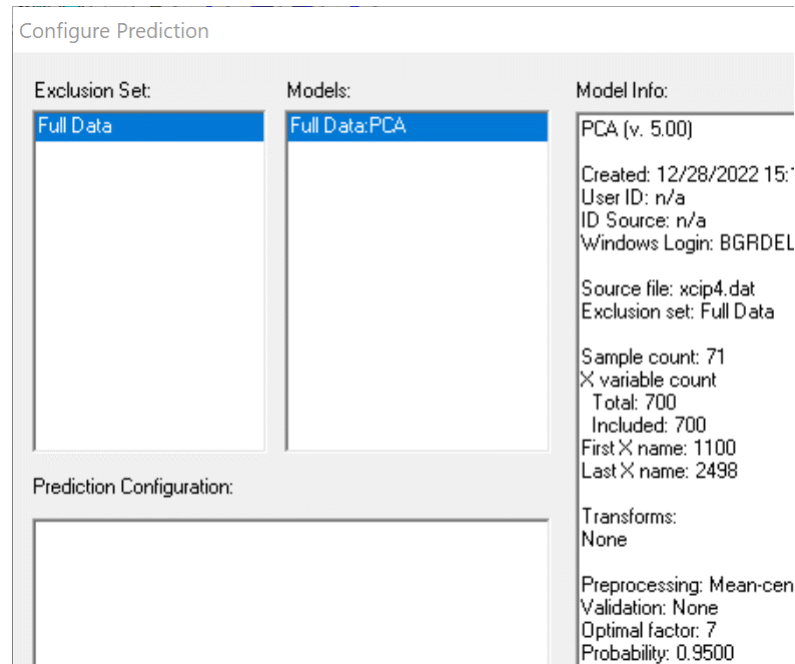
		1	2	3
b		Variance	Percent	Cumulative
1	Factor1	7974515.5	55.148037	55.148037
2	Factor2	6170441.0	42.671898	97.819931
3	Factor3	170377.21	1.178250	98.998184
4	Factor4	67367.203	0.465880	99.464066
5	Factor5	59778.425	0.413400	99.877464
6	Factor6	6060.7500	0.041913	99.919380
7	Factor7	4812.2758	0.033279	99.952660
8	Factor8	3506.0935	0.024247	99.976906
9	Factor9	2048.9936	0.014170	99.991074
10	Factor10	1289.2429	0.008916	99.99992

MAKING A PCA PREDICTION

Running PCA triggers the creation of a model. You can confirm this by going to Process/Predict after running the algorithm and noting the entry under Model. Making predictions requires a model and a target, that is, a data set with an x block containing one or more samples that are often referred to as *unknowns*. This data set's x block must have the same number of independent variables as the data set from which the model was created and cannot contain any excluded independent variables. The prediction target may or may not contain dependent and class variables.

Figure 5.43 shows the Configure Prediction dialog box. To get model information, highlight its entry as illustrated in the figure. To configure a prediction, highlight a model and exclusion set and click on Add. You can configure more than one prediction at a time by highlighting a different model name or exclusion set and again clicking Add. Predictions are made when you click Run.

Figure 5.43
Configure Prediction
dialog box



Factor Select

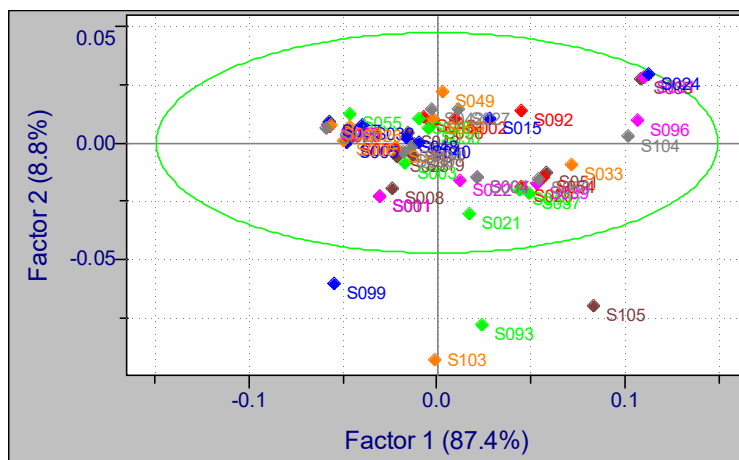
The Factor Select object displayed after prediction contains the first three columns of the training set Factor Select object. It mainly provides a mechanism for changing the number of model factors during prediction. Initially, the diamond cursor is set at the value determined during the modeling phase. However, the user can change the setting and see the effect on objects dependent on the number of model factors.

Normally, you would not change the diamond from the initial position defined when creating the model. However, depending on the nature of the prediction data, you may want to evaluate the effect of the model with fewer or more factors.

Scores

The Scores object produced during prediction includes a confidence threshold when either a 2D or Multiplot view is displayed. The location of the ellipse depends on the probability level set in Windows > Preference > Prediction. [Figure 5.44](#) shows an example of a confidence ellipse drawn on a 2D scatter plot.

Figure 5.44
Scores plot with a
confidence ellipse

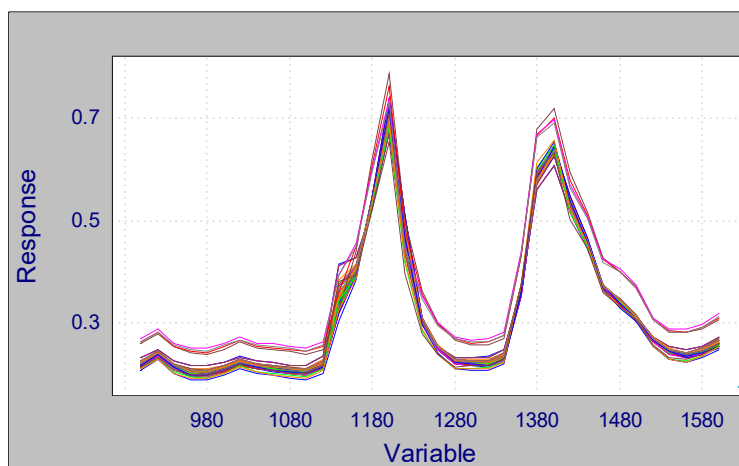


If an unknown's predicted score lies outside the thresholds, it is somehow unlike the training set samples. You should examine the X Residuals and the Outlier Diagnostics of samples which lie well beyond the confidence ellipse.

X Reconstructed

For a discussion of this object, see "X Reconstructed" on page 5-37.

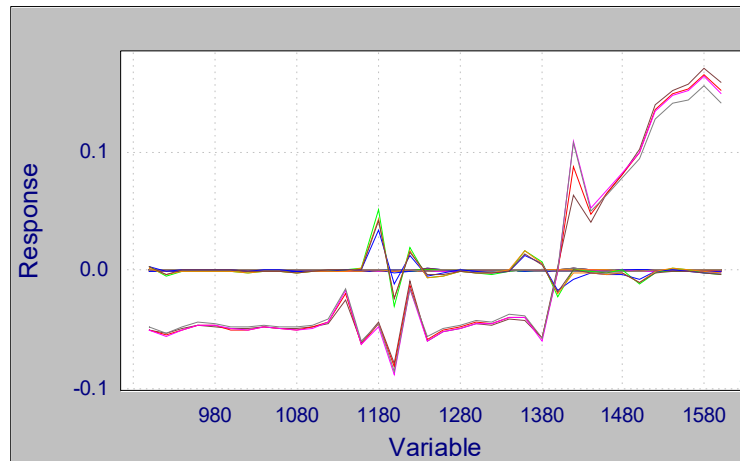
Figure 5.45
PCA Predict X
Reconstructed
object



X Residuals

For an introduction to this object, see "X Residuals" on page 5-38. Unknowns similar to the training set have residual values, for all variables, of comparable magnitude and structure. However, unknowns differing significantly from the training set (*i.e.*, outliers) will have residual values that deviate notably at certain variables, as shown in the figure below. Compare Figure 5.35 to Figure 5.46, paying special attention to the difference in the y axis scale. In this case, the X Residuals structure of prediction outliers indicates what is missing from the training set but present in the outliers. This information can be extremely helpful when trying to decide in what way an outlier is different. For example, in molecular and mass spectrometry, the residual spectrum might be used to identify a contaminant compound.

Figure 5.46
Prediction X
Residuals

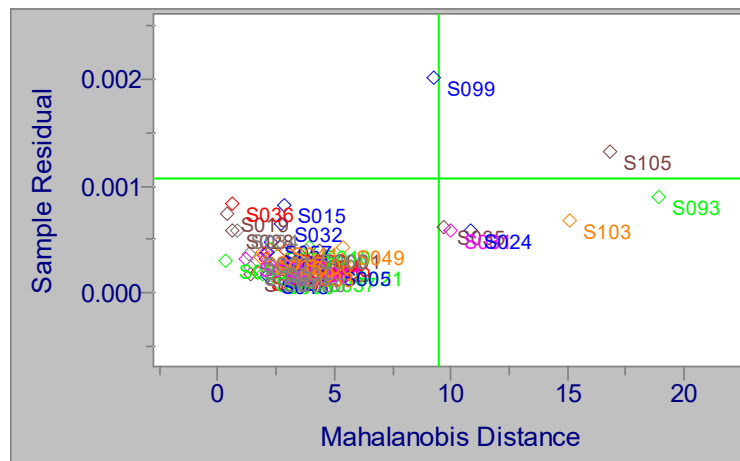


The square of the X Residuals are equivalent to Error Contributions, useful for an understanding of where among the variables a unusual sample differs from those in a model. See “Contributions” on page 5-40 for more discussion of contribution plots.

Outlier Diagnostics

This object contains the same three quantities described in “Outlier Diagnostics” on page 5-38. Note, however, that during prediction, the augmented sample residual defined by equation 5.46 (not equation 5.25), the Mahalanobis Distance defined in equation 5.48 (not equation 5.30), and the probability derived from equation 5.47 (not equation 5.27) are computed.

Figure 5.47
Prediction Outlier
Diagnostics



Despite their small differences in computation, prediction outlier diagnostics should be interpreted in the same manner as those developed during calibration, that is, samples which lay beyond the suggested thresholds should be considered possible outliers.

References

1. Hartigan, J.A.; *Clustering Algorithms* (Wiley: New York, 1975).

2. Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; and Vetterling, W.T.; *Numerical Recipes* (Cambridge: Cambridge University Press, 1986), pp. 52-64.
3. Miller, P., Swanson, R.E. and Heckler, C.E. "Contribution Plots: A Missing Link in Multivariate Quality Control", *Appl. Math. And Comp. Sci.* (1998) 8(4): 775-792.
4. Kourti, T. and MacGregor, J.F. "Multivariate SPC Methods for Process and Product Monitoring", *J. Quality Technology*, (1996) 28:409-428.
5. Malinowski, E.R.; *Factor Analysis in Chemistry*, 2nd Edition (Wiley: New York, 1991), p. 98.
6. Henry, R.C.; Park, E.S., and Spiegelman, C.H. "Comparing a new algorithm with the classical methods for estimating the number of factors". *Chemometrics Intell. Lab. Systems.* (1999) 48:91-97.
7. Jackson, J.E. and Mudholkar, G.S. "Control Procedures for Residuals Associated with Principal Component Analysis", *Technometrics* (1979) 21:341-349.
8. http://en.wikipedia.org/wiki/Mahalanobis_distance
9. Haaland, D.M. and Thomas, E.V.; "Partial Least-Squares Methods for Spectral Analysis. 1. Relation to Other Quantitative Calibration Methods and the Extraction of Qualitative Information", *Anal. Chem.*, (1988) 60:1193-1202.
10. Osten, D.W.; "Selection of Optimal Regression Models via Cross-Validation", *J. Chemometrics*, (1988) 2: 39.
11. Wold, H.; "Estimation of Principal Components and Related Models by Iterative Least Squares", in Krishnaiah, P.R., Ed., *Multivariate Analysis, Proc. International Symposium, June 1965* (Academic Press: New York, 1966) pp. 391-420.
12. Geladi, P.; "Partial Least Squares Regression: A Tutorial", *Anal. Chim. Acta* (1986) 185:1-17.
13. Vandeginste, B.G.M.; Sielhorst, C. and Gerritsen, M.; "The NIPALS algorithm for the calculation of the principal components of a matrix", *TrAC, Trends Anal. Chem.* (1988) 7 (8): 286-287.
14. Miyashita, Y.; Itozawa, T.; Katsumi, H. and Sasaki, S.; "Short Communication Comments on the Nipals Algorithm", *J. Chemometr.* (1990) 4: 97-100.
15. Harmon, H.H.; *Modern Factor Analysis* (University of Chicago Press: Chicago, 1976) Chapters 12-13.
16. Forina, M.; Lanteri, S. and Leardi, R.; "A program for varimax rotation in factor analysis" *TrAC, Trends Anal. Chem.*, (1987) 6 (10): 250-251.
17. ARTHUR, Version 4.1 (1984) Infometrix, Inc.

Reading List

1. Beebe, K.R.; Pell, R.J.; and Seasholtz, M.B.; *Chemometrics: A Practical Guide*, (John Wiley & Sons: New York, 1998).

5 Exploratory Analysis: References

2. Massart, D.L.; Vandeginste, B.G.M.; Deming, S.N.; Michotte, Y.; and Kaufman, L; *Chemometrics: a textbook*, (Elsevier: Amsterdam, 1988).
3. Malinowski, E.R.; *Factor Analysis in Chemistry*. Second Edition, (John Wiley & Sons: New York, 1991).
4. Manley, B.F.J.; *Multivariate Statistical Methods: A Primer*, (Chapman and Hall: London, 1986).
5. Jackson, J.E.; *A User's Guide to Principal Components*, (John Wiley & Sons: New York, 1991).
6. Montgomery, D.C.; *Introduction to Statistical Quality Control*, (John Wiley & Sons: New York, 2005).

Classification Methods

Contents

K Nearest Neighbors	6-2
Soft Independent Modeling of Class Analogy	6-15
Calibration Transfer	6-29
References	6-30

In many technical fields, it is not uncommon to accumulate large quantities of data comprising a variety of measurements with the ultimate goal of substituting an easy determination for a difficult one. Here the term *difficult* may imply expensive, time-consuming, or inconvenient while the term *easy* may imply automated. If the difficult determination is the prediction of a continuous property, the techniques discussed in [Chapter 7, Regression Methods](#) are appropriate. However, many problems can be couched in terms of category prediction. For example, suppose you want to decide if a product is in or out of spec from its UV-vis spectrum. Such a problem requires constructing a classification model which establishes a relationship between a product's spectrum and its acceptability. Pirouette's approach to classification modeling is described in this chapter.

Classification models can be based on either probability, separability or similarity. The probabilistic approach assumes that: (1) measured values for like samples tend toward a uniform distribution; (2) measurements on an unknown fall within the allowable distributions of samples belonging to the same class as the unknown; and (3) measurements on an unknown can be distinguished from the allowable distributions of values for samples belonging to other classes. The most well-known instance of probabilistic classification derives from Bayes Theorem¹. Bayesian methods are very powerful in sample-rich scenarios, especially where there are many samples in each category and more samples than variables. The more samples, the better the approximation of the true distributions by the observed distributions. However, probabilistic methods applied to sample-poor/variable-rich data often produce models said to be overfit. Overfit models make poor predictions. For this reason Bayesian methods are not included in Pirouette.

Separability approaches, also not included in Pirouette, assume that groups can be distinguished by finding gaps between them in the measurement space. Some of the earliest pattern recognition methods are of this type, including the Linear Learning Machine² (LLM) and Linear Discriminant Analysis³ (LDA). All separability techniques tend toward overfitting when variables outnumber samples. In the past, this was irrelevant; most data sets were sample-rich because making measurements was such a time-consuming endeavor. Today, however, spectroscopic and chromatographic methods routinely generate hundreds of measurements per sample. In this variable-rich world over-fitting is a real concern. Methods based on separability have an additional weakness: instability.

The equation for the plane separating two categories depends on sample order; changing that order changes the solution. Finally, neither LDA nor LLM can handle the so-called asymmetric case, where one category (*e.g.*, good product) is surrounded in multivariate space by another category (*e.g.*, bad product).

Similarity techniques are based on the assumption that the closer samples lie in measurement space, the more likely they belong to the same category. This idea of proximity implies the concept of distance. Pirouette's two classification algorithms, K-Nearest Neighbor (KNN) and Soft Independent Modeling of Class Analogy (SIMCA), are similarity techniques which differ in their distance definition. Both KNN and SIMCA construct models using samples preassigned to a category, *i.e.*, a supervised pattern recognition approach. Usually these assignments are derived from knowledge external to the independent variable measurement. Sometimes, however, categories must be defined based on clusters found during an exploratory analysis because external information is not available, *i.e.*, an unsupervised pattern recognition approach. Both algorithms are implemented in two stages. First, a model is built and refined based on a training set (*i.e.*, the knows); later it is used to predict classes of new samples (*i.e.*, the unknowns).

The two methods are complementary in many respects. KNN is well-suited to a sample poor environment; it can function even with only one training set sample per category and performs adequately when categories are sub-grouped. It is simple to understand and the model refinement phase can be brief since KNN provides few diagnostics. A KNN prediction consists of assigning each unknown to one and only one category defined in the training set. In contrast, SIMCA requires that each training set category be a homogeneous group of several samples, and model refinement and prediction are much more complicated. Many diagnostics are available and examining all can be time-consuming. The payoff is a detailed picture of model structure and more realistic prediction options. An unknown in SIMCA can be assigned to more than one category and a probability for each assignment is calculated. Moreover, an unknown may be deemed to belong to none of the categories included in the training set.

The remainder of this chapter is devoted to a description of Pirouette's two most reliable classification algorithms. The mathematical basis of each is given and the computed objects detailed.

Both of these algorithms require that a class variable be activated (see [“Activating a Class Variable” on page 13-19](#)). If only one class variable is present in the file, that variable will be activated upon opening the file. However, if more than one class variable is present, you should choose which will be used by the classification algorithm.

Beginning with version 4.0, Pirouette enables the use of another algorithm for performing classification, PLS Discriminant Analysis. PLS-DA uses the PLS algorithm to build regression models correlating the information in the X block to binary Y variables. For details on this algorithm and its use, see [“PLS for Classification” on page 7-38](#).

K Nearest Neighbors

KNN attempts to categorize an unknown based on its proximity to samples already placed in categories⁴. Specifically, the predicted class of an unknown depends on the class of its k nearest neighbors, which accounts for the name of the technique. In a fashion analogous to polling, each of the k closest training set samples votes once for its class; the unknown is then assigned to the class with the most votes. An important part of the process is determining an appropriate value for k, the number of neighbors polled.

MATHEMATICAL BACKGROUND

The multivariate distance used in KNN is similar to the distance separating two points in a plane, but with N coordinates in the calculation, rather than two. The general expression for this Euclidean distance, d_{ab} , is:

$$d_{ab} = \left[\sum_{j=1}^m (a_j - b_j)^2 \right]^{1/2} \quad [6.1]$$

where \mathbf{a} and \mathbf{b} are the data vectors for the two samples. A data vector contains the m independent variable measurements made on each sample. HCA, discussed in [Chapter 5, Exploratory Analysis](#), is based on this same Euclidean distance.

In the KNN model building phase, the Euclidean distance separating each pair of samples in the training set is calculated from [equation 6.1](#) and stored in a distance table. For any particular sample, the classes of its nearest neighbors can be tallied and the sample assigned to the class to which most of the nearest neighbors belong. If two (or more) classes get the most votes, the tie is broken based on accumulated distances, *i.e.*, distances are summed instead of votes. The sample in question is then considered to belong to the class with smallest accumulated distances.

To distinguish between the *a priori* category assignment and the assignment inferred from the KNN algorithm, we use the terms m-class (for measured) and p-class (for predicted). During model building, we look for m-class and p-class agreement. Substantial agreement suggests that the *a priori* assignments are reasonable and that sample similarity is embodied by the independent variable responses. Where disagreements exist, they may be caused by outliers: samples either with incorrect m-class assignments or associated with flawed independent variable measurements. Disagreements may also arise from an inappropriate k value. The value of k can range from 1 to one less than the total number of samples in the training set. However, when k approaches the size of the training set, the k th nearest neighbor is actually a far away neighbor.

The following simple example illustrates some key ideas in KNN.

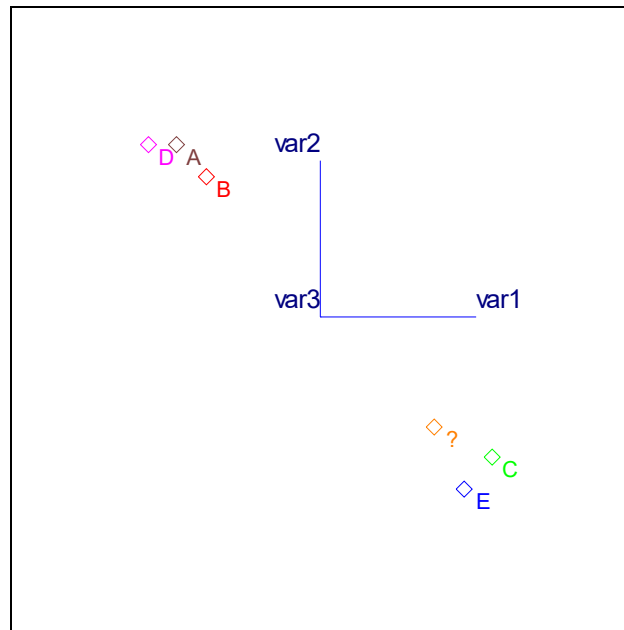
NEAREST NEIGHBOR EXAMPLE

[Table 6.1](#) contains a small data set which is plotted in [Figure 6.1](#). The data set contains five training set samples (A through E) and one unknown (?). Three measurements have been made on the five samples.

Table 6.1
Data for five training
set samples and an
unknown

	var1	var2	var3
A	1.0	2.0	3.0
B	1.1	1.9	2.9
C	2.1	1.0	2.4
D	0.9	2.0	3.0
E	2.0	0.9	2.3
?	1.9	1.1	1.8

Figure 6.1
3D plot of [Table 6.1](#)
data



It is clear from looking at the raw data in [Table 6.1](#) that A, B and D resemble each other more than either resembles C or E. The 3D plot in [Figure 6.1](#) emphasizes that point; samples A, B and D are yellow while C and E are red. A three-variable example allows us to completely visualize the relationships between samples. [Table 6.2](#) contains the intersample distances. Because the table is symmetrical about the diagonal, only half the entries are shown.

Table 6.2
Intersample
distances for
[Table 6.1](#) data

	A	B	C	D	E	?
A	0	0.173	1.603	0.100	1.643	1.749
B		0	1.435	0.245	1.473	1.578
C			0	1.673	0.173	0.640
D				0	1.706	1.803
E					0	0.548
?						0

We can see from both [Figure 6.1](#) and the last column in [Table 6.2](#) that the unknown is, in fact, closest to C and E.

Suppose we now supply the additional information which must be included in a training set, the *a priori* m-class assignments. Let's assume that A, B and D belong to Category 1 while C and E belong to Category 2. Part of the modeling phase consists of deciding whether to poll 1, 2, 3, or 4 neighbors. This can be accomplished by looking for p-class and m-class agreements in the training set.

If we poll only the nearest neighbor (i.e., $k=1$), the p-class and m-class of all training set samples coincide. If two neighbors are polled, the agreement persists. Polling results for $k=2$ are **always** identical to that for $k=1$ because of the KNN tie-breaker mechanism. If the nearest neighbor is of a different class than that of the second nearest neighbor, the p-class of a test sample is based on the actual distances to the samples casting the tying votes. A 1 to 1 tie is resolved by choosing the class of the first nearest neighbor; by definition, it has the shorter distance to the test sample.

If the three nearest neighbors of each sample are polled, two misses occur. Both C and E, which are assumed to belong to category 2, have p-classes of 1 since two of their three nearest neighbors vote for class 1. If four nearest neighbors are polled, the same two misses occur. If we decide to tolerate no misclassifications in the training set, k should be fixed at 2.

If the category of the unknown is predicted with $k = 2$, sample “?” is assigned to Category 2 because the two nearest training set samples, C and E, both belong to that category.

RUNNING KNN

Before running KNN, it is necessary to activate a class variable; for instructions and for the rules governing class variables, see “[Class Variables](#)” on page 13-19. Then, to configure a KNN run, the user must select preprocessing options and transforms and specify a maximum k value, that is, the largest number of neighbors to be polled, which is denoted k_{\max} . A rule of thumb for this setting is less than twice the size of your smallest category. When the KNN algorithm is run, the number of neighbors polled ranges from 1 to k_{\max} , and each sample is classified in this way for every value of k . The Configure Run parameters for KNN are shown below.

Figure 6.2
Configuring a KNN
run

The image shows a software dialog box titled "K-Nearest Neighbor". It contains the following elements:

- Preprocessing:** A dropdown menu with "Autoscale" selected.
- Maximum Neighbors:** A text input field containing the number "10", followed by a range indicator "(1-74)".
- Enable Calibration Transfer:** A checkbox at the bottom left, which is currently unchecked.

The six objects produced by KNN are described in this section. What information they contain and how they should be viewed and manipulated is detailed. In addition to the computed objects, a model containing information necessary to make predictions is created. This model is automatically stored along with the KNN objects. It can also be stored in a file separate from the training set data and algorithm results and reloaded later to make predictions on future samples. A Pirouette KNN model contains information about which variables were excluded and what transforms/preprocessing options were chosen so that future samples are treated in the same way as the training set. Model building is an iterative process. You seldom run the KNN algorithm just once and immediately start making predictions. Instead you spend most of your time finding the “best” set of samples, variables and algorithm options; see “[Optimizing the Model](#)” on page 6-10.

X Preprocessed

Preprocessing will have an impact on the quality of the model you produce. This object allows you to visualize the effect on the raw data and allows you to evaluate sections of

the data range that may not have much impact on the classification model because of either invariance or randomness: see “X Preprocessed” on page 5-32

Votes

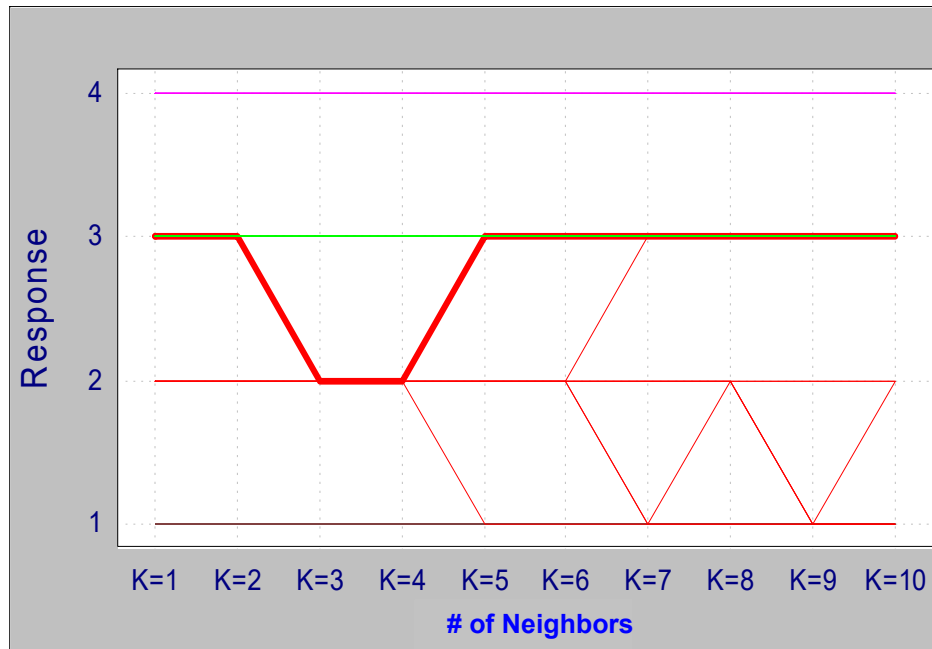
This object holds the basic result of a KNN classification: the p-class for each sample at each k value. Votes is a matrix having n rows and k_{max} columns, where n is the number of training set samples. The first column of this matrix holds the p-class for each training set sample when only one neighbor (the nearest) is polled. The last column holds the p-class for the samples when the k_{max} nearest neighbors are polled. Thus, column number corresponds to k setting. Figure 6.3 shows Votes for the example training set shown in Figure 6.1. When k=2, sample 17 is placed in category 3; however, when k=3, it is placed in category 2.

Figure 6.3
Votes

		1	2	3	4	5	6
		K=1	K=2	K=3	K=4	K=5	K=6
10	101K-1A	1	1	1	1	1	1
11	212BLAV1	2	2	2	2	2	2
12	312BLAV9	2	2	2	2	2	2
13	202BL-2	2	2	2	2	2	2
14	302BL-3	2	2	2	2	2	2
15	502BL-6	2	2	2	2	1	1
16	602BL-7	2	2	2	2	2	2
17	112BLAV7	3	3	2	2	3	3
18	102BL-1	2	2	2	2	2	2
19	702BL-8	2	2	2	2	2	2
20	103SH-1	3	3	3	3	3	3
21	203SH-15	3	3	3	3	3	3

The Votes object can be a little overwhelming in the table view if the number of samples and/or categories is large. A line plot view with sample # on the x axis and p-class on the y axis simplifies things considerably. Figure 6.4 shows such a line plot.

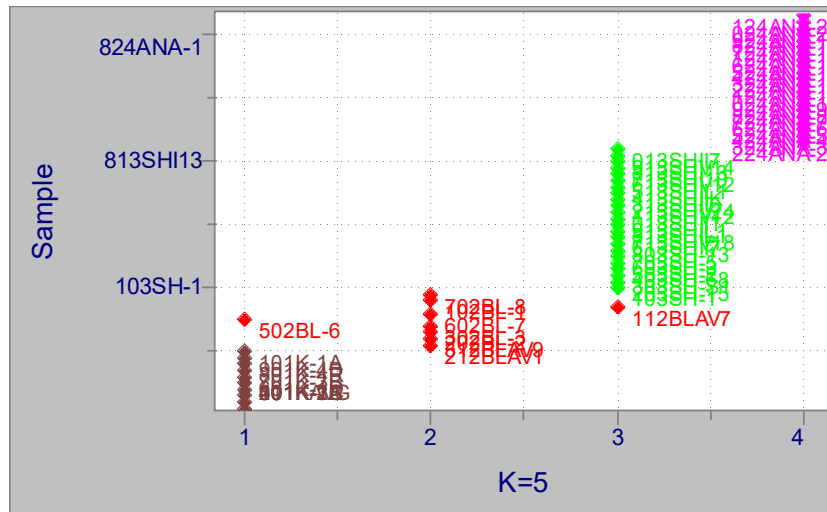
Figure 6.4
Line plot of Votes;
red trace for K=1,
green for K=3



A single trace means that every sample's p-class does not change with k. This is observed only when categories are extremely well-separated. More often, multiple traces with peaks or dips occur for samples whose p-class varies with the number of neighbors polled. The line plot of Votes shown in Figure 6.4 confirms that the highlighted sample changes from category 3 to category 2 when k is changed from 2 to 3 neighbors.

Another view of Votes tracks misclassifications for a particular number of nearest neighbors. A 2D scatter plot with sample # on the y axis and the k value of interest is very effective in pointing out misclassified samples if the class variable has been activated. Such a plot is shown in Figure 6.5.

Figure 6.5
Votes as a 2D plot



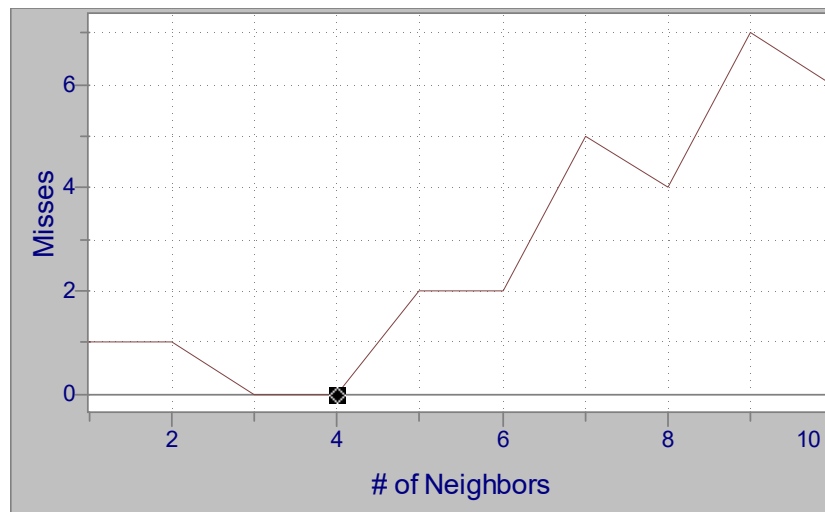
Here, the results from using 5 neighbors are plotted. For 3 of the categories, no samples are misclassified. Category 1 samples (in brown) stack vertically, with no brown-labelled samples elsewhere in the plot. Similar observations can be made regarding categories 3

and 4. However, two Category 2 samples (in red) have been misclassified. Plotting in this manner quickly pinpoints classification errors.

Total Misses

If Votes entries (p-classes) are compared to the *a priori* assignments (the m-classes), a more informative object, Total Misses, is created. It is generated by first replacing Votes entries with zero if the m-class and p-class coincide and by one if they differ and then summing down the columns. It indicates the total number of misclassifications (or misses) at each k setting. As such, it is key in determining the optimal number of neighbors to be polled. The Total Misses corresponding to the Votes matrix of Figure 6.3 is shown in Figure 6.6.

Figure 6.6
Total Misses with the diamond at k=1.



The diamond at k=4 indicates the current setting for the optimal number of neighbors. Pirouette always initially positions the diamond at the minimum k which yields the minimum misses. You can move the diamond to another k setting by clicking the left mouse button above the desired k value on the trace.

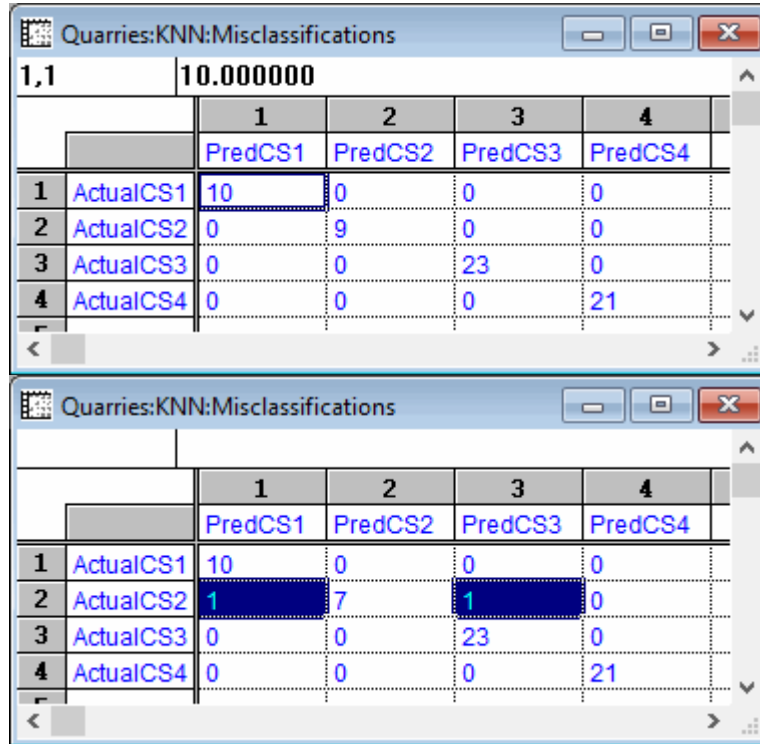
When k is much larger than the smallest training set class, the total number of misses tends to increase. To understand why, consider a test sample belonging to the smallest m-class and located near the other samples in that class. When k is less than the number of samples in the class, the test sample receives votes mostly from its fellow m-class members. However, as more neighbors are polled, the votes of samples belonging to another (presumably more distant) m-class eventually predominate and the test sample is misclassified.

Misclassifications and Total Misses

The third KNN object, Misclassifications, summarizes classification success by category. For a given k setting, a square table is constructed with as many rows and columns as training set classes. The row labels—MeasN—correspond to the training set m-classes, and the column labels—PredN—correspond to the p-classes. The entries along the descending diagonal of the table indicate the number of samples correctly classified; off-diagonal entries are the number of samples misclassified. The Misclassifications entries depend on the diamond setting in Total Misses. When the diamond is moved, this object is recalculated and the values may change.

Two Misclassifications objects are shown in Figure 6.7 in the table view. The upper corresponds to k=4 while the lower corresponds to k=5. Note that the sum of the off-diagonal elements in this object must equal the Total Misses for the corresponding k setting.

Figure 6.7
Misclassifications
objects for:
(a) k = 4, (b) k = 5



Class Fit and Fit Thresholds

KNN classifies every sample into the closest training set m-class but the user is left wondering if that category is close enough, *i.e.*, is the assigned category a reasonable conclusion. To qualify the prediction result, the Class Fit (or “goodness”)⁵, has been incorporated into Pirouette. The basic idea is to compute the distance of the sample to the nearest member of its predicted class and then compare that value to some distance which estimates class “diameter”. If the distance to the class is significantly larger than the class “diameter”, the likelihood the sample belongs to that class decreases. The estimate of class “diameter” is the average of the smallest intersample distances for each training set sample in the class.

Thus, Class Fit, g_i is computed from

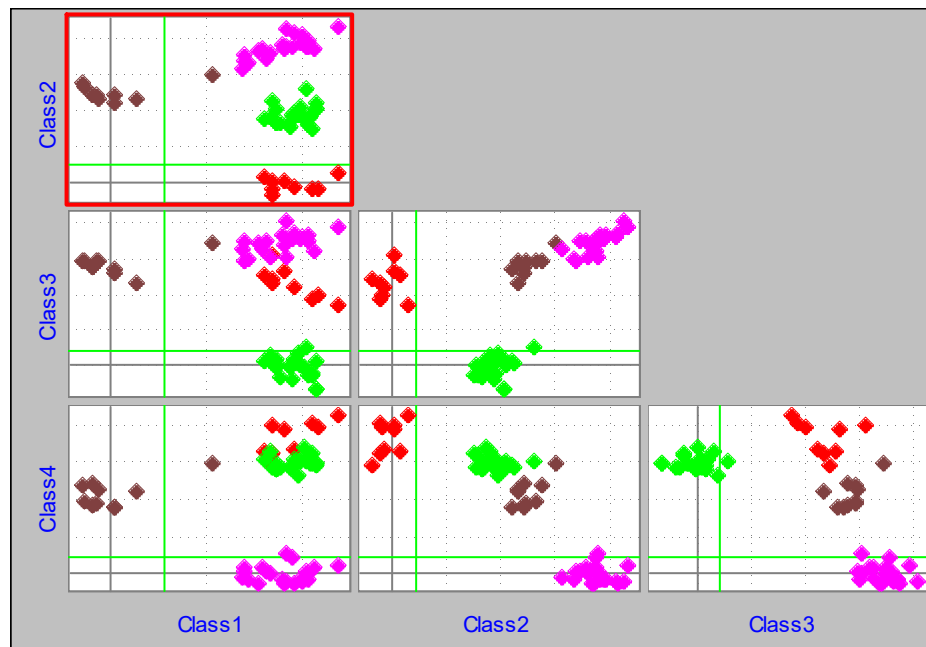
$$g_i = \frac{d_i - \bar{\mathbf{d}}_q}{sd(\mathbf{d}_q)} \tag{6.2}$$

where d_i is the distance of the sample to the nearest member of the class, and \mathbf{d}_q is a vector containing the smallest distances of each category member to all category members. The denominator contains the standard deviation of \mathbf{d}_q .

Class Fit is calculated for every sample against all classes. The resulting matrix is most easily displayed as a multiplot as shown below. The metric, similar to a t value, indicates the number of standard deviation units the sample’s distance is from an average (*i.e.*, ex-

pected) distance in the class. Thus, the plot has a threshold derived from the t distribution with $\alpha = 0.05$ and degrees of freedom equal to the number of samples in the class.

Figure 6.8
Class Fit object



Interpretation of a Class Fit plot is like that of a decision diagram. Samples laying to the left of the vertical threshold qualify as members of the category on the x-axis, while samples below the horizontal line would be members of the category on the y-axis.

The Class Fit Thresholds object is simply a list of the values for the t distribution thresholds for each class in the dataset.

Class Fit can bolster confidence in a good classification but can also flag possible outliers. Values smaller than the threshold are a good indication that the sample qualifies as a class member. Negative values occur when the sample's distance is smaller than the class average. Because the threshold line is determined from a distributional statistic, 5% of the population is expected to exceed it. Control over the statistic is possible in KNN predictions; see "Class Fit" on page 6-14.

Do not examine Class Fit plots to determine predicted category. A sample may have a very small Class Fit value for a category it does not belong to. Remember that Class Fit is based on the distance to the single nearest neighbor; the predicted class, however, may be based on $k > 1$.

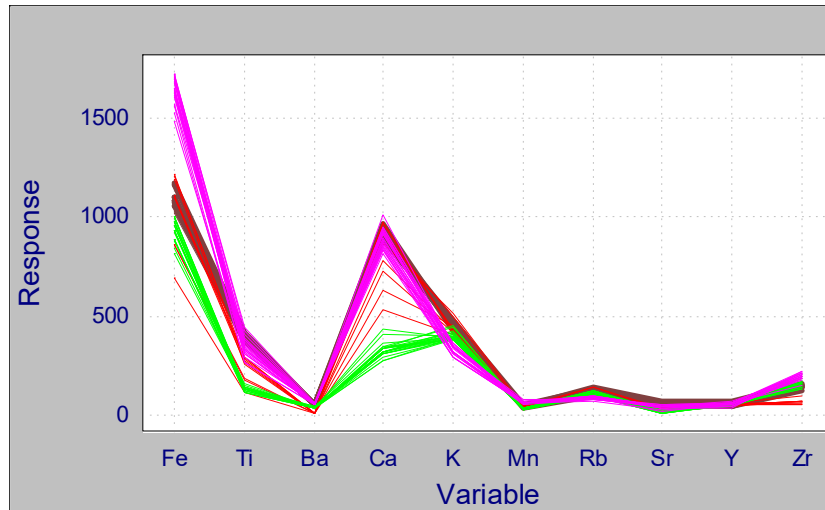
Note: The Class Fit will not be computed if there are fewer than 4 samples in a category.

OPTIMIZING THE MODEL

The model must be optimized after the algorithm runs. This requires determining an appropriate value of k , the number of neighbors polled. It also may be advantageous to find and remove gross outliers from the training set, particularly if the number of samples per category is small. In addition, you may want to exclude variables which do not distinguish insufficiently separated categories.

This last process is the most straightforward. Because the KNN algorithm yields no variable-based diagnostics, decisions about excluding variables must be made by examining line plots of the original data. Figure 6.9 contains a line plot of the data used in the previous discussion, which demonstrates that at least the first 4 or 5 variables are helpful in distinguishing among the categories. For large data sets, activating the class variable before (or after) plotting greatly simplifies the task of deciding if classes can be distinguished by a variable or variable region, based on their line colors. See “Activating a Class Variable” on page 13-19 for a discussion of the Activate Class function, which maps color to samples based on class.

Figure 6.9
Line plot of example
data set



Choosing an appropriate k setting and identifying outliers are related tasks. If k is too small and classes overlap or outliers are nearby, a sample’s p -class may be unduly influenced. If k is too large, a sample’s p -class is skewed toward large nearby classes.

Before choosing a value of k , you should know how many categories are in your training set and the size of each. This, along with the nature of the problem you seek to solve, should give you an idea of how many total misclassifications are acceptable. For example, you may feel comfortable if 95% of your samples classify correctly in your optimized model and if no class contains more than one misclassification.

A general strategy for determining k involves inspecting the Misclassification object at various k settings. Check to see if misses are concentrated in two classes. For example, suppose that most of your misses are in classes 3 and 7. If Misclassification indicates that class 3 misses predict into class 7 and that class 7 misses predict into class 3, this is evidence of class overlap. Detect this problem during the model building phase so you can view later predictions for overlapped classes with some skepticism.

If misses are distributed over many classes and if the same samples misclassify at different k values, this is consistent with outlier behavior. It is a good idea to compare line plots of a potential outlier and samples in both its m -class and p -class before excluding it. Perhaps the outlier has the wrong m -class, that is, it actually belongs to a different class in

the training set. Fortunately, when the number of outliers is small compared to the class size and to the optimal k , their influence is negligible.

Note: *When the training set has a very small number of samples, you may be forced to choose 1 for the optimal number of neighbors. Such a low value is usually discouraged because of the greater chance of a misleading prediction when classes are not well-separated or contain outliers. A value of k between 3 and 5 is preferable.*

After excluding samples or variables, you **must** rerun the KNN algorithm. Moreover, if you are not satisfied with the number of misses in the optimized model resulting from the preprocessing and transforms set initially, you may want to investigate other configurations (e.g., “Transforms”).

HCA AS A KNN VIEWING TOOL

KNN and HCA (discussed in [Chapter 5, Exploratory Analysis](#)) are very closely related; both start by calculating multivariate distances for pairs of samples. The Single Linkage dendrogram is a graphical representation of a KNN model with $k=1$. Thus, it can indicate whether a particular data set is likely to produce a viable KNN model. Compare the m -classes to the class variable created when you activate a class from the dendrogram view; see “[Creating Class Variables](#)” in [Chapter 12](#). The similarity of those two listings of class variables provides a form of diagnostic of modeling potential.

MAKING A KNN PREDICTION

When you have optimized your KNN model, you are ready to make predictions with it. To see a list of all loaded models of all types, go to Process/Predict. Click on an entry under Model to display information about it. [Figure 6.10](#) shows a typical KNN model entry and its associated information.

Figure 6.10
Configure Prediction
dialog box

Exclusion Set:	Models:	Model Info:
Full Data Quarries Artifacts	Quarries:KNN	KNN (v. 5.00) Created: 12/29/2022 10:12 User ID: n/a ID Source: n/a Windows Login: BGRDELL Source file: ARCH.xls Exclusion set: Quarries Sample count: 1 (10), 2 (9), 3 (23) 4 (21) X variable count Total: 10 Included: 10 First X name: Fe Last X name: Zr Transforms: None Preprocessing: Autoscale Validation: None Class Variable: Quarry Optimal neighbors: 1

To make a KNN prediction, go to Process/Predict and select a model and a target exclusion subset. You can configure several predictions at once. Clicking on Run triggers the predictions.

Note: *Grayed exclusion sets contain excluded variables and cannot be configured.*

Class Predicted

The predicted class of each sample in the prediction target exclusion set is tabulated in the Class Predicted object. Note that every sample in the target exclusion set has one and only one p-class. The predicted category of the first 10 samples are highlighted in Figure 6.11.

Figure 6.11
Class Predicted

Full Data:KNN-2 Predict:Class Predicted		10,1 1.000000			
		1	2	3	
		Class			
1	111KAVG	1			
2	201K-1B	1			
3	301K-2	1			
4	401K-3A	1			
5	501K-1C	1			
6	601K-1D	1			
7	701K-3B	1			
8	801K-4R	1			
9	901K-4B	1			
10	101K-1A	1			
11	212BLAV1	2			

Misclassification Matrix

The prediction Misclassifications object differs from the modeling analog in that the table contains an additional line—Unmodeled—which shows the predicted category for those samples not assigned an *a priori* category value or assigned to a category not present in the model. An example result follows.

Figure 6.12
KNN
Misclassifications
during prediction

		1	2	3	4
		PredCS1	PredCS2	PredCS3	PredCS4
1	ActualCS1	10	0	0	0
2	ActualCS2	0	9	0	0
3	ActualCS3	0	0	23	0
4	ActualCS4	0	0	0	21
5	Unmodele	3	3	6	0

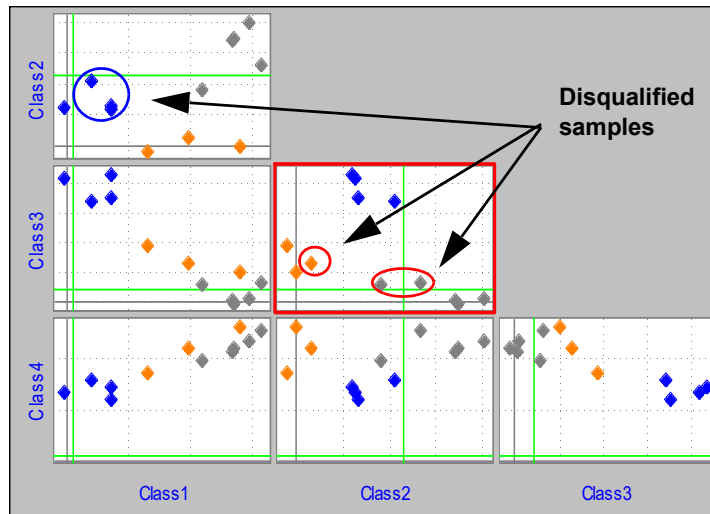
Note: *The Misclassification Object will be computed only if there is a Class variable in the prediction set which has the same name as the Active Class used during the modeling phase. The name match is case sensitive.*

In the example above there are 12 samples with categories not represented in the training set model; they are summarized in the last row labeled Unmodeled. These samples are still classified, however.

Class Fit

The Class Fit object produced during KNN prediction has the same utility as that described in “[Class Fit and Fit Thresholds](#)” on page 6-9. One difference is that the user can set the Probability level associated with the threshold. Use the values to confirm the KNN prediction or, if the value falls beyond the threshold, consider that the sample may not belong to the predicted class.

Figure 6.13
KNN Prediction
Class Fit object, with
5 objects
disqualified (circled)



Note: Training set samples also present in a prediction set will have Class Fit values equal to the negative of the average of the minimum class distances divided by the standard deviation of the minimum class distances. Because these samples are also in the model, the distance to the nearest sample is 0, each sample being its own nearest neighbor.

Soft Independent Modeling of Class Analogy

The Soft Independent Modeling of Class Analogy (SIMCA) method was first introduced by Svante Wold in 1974⁶. Since that time the acronym has changed, but the functionality of the method has been demonstrated and enhancements offered by a number of researchers⁷. In contrast to KNN, which is based on distances between pairs of samples, SIMCA develops principal component models for each training set category. Later, when the x block of a new sample is projected into the PC space of each class, the new sample is assigned to the class(es) it best fits.

Reliable classification of new samples (*i.e.*, unknowns) is the ultimate goal of SIMCA. However, the technique also provides a rich set of diagnostics which address other interesting aspects of classification. For example, modeling power points out the most important variables in the training set, and Mahalanobis distance provides a probabilistic means of identifying and ranking outliers. Moreover, the variance structure of each class yields clues about category complexity and may even reveal the phenomena which cause one category to differ from another.

A very attractive feature of SIMCA is its realistic prediction options compared to KNN. Recall that KNN assigns every sample to exactly one training set class, the so-called nearest neighboring class, even though this class may not be near in any absolute sense. SIMCA, however, provides three possible prediction outcomes:

- The sample fits only one pre-defined category
- The sample does not fit any pre-defined categories
- The sample fits into more than one pre-defined category

In addition, because these decisions are made on the basis of statistical tests, we can express outcomes probabilistically.

MATHEMATICAL BACKGROUND

In SIMCA, each training set class is described by its own principal component (PC) model. The underlying mathematics is discussed in “Principal Component Analysis” in Chapter 5. The material in “Mathematical Background” on page 5-16 up to “NIPALS” on page 5-28 is pertinent to SIMCA, and the reader is strongly encouraged to review it to become generally familiar with PCA modeling and its various results (*e.g.*, loadings, scores, modeling power, sample residual, *etc.*). When a prediction is made in SIMCA, new samples insufficiently close to the PC space of a class are considered non-members. This approach is similar to that developed in “Predicting in PCA” on page 5-29. Additionally, SIMCA requires that each training sample be pre-assigned to one of Q different categories, where Q is typically greater than one. Thus, SIMCA can be considered a Q -class PCA with $Q > 0$.

Four important topics remain: (1) the between class measures, that is, how class separation is quantified when $Q > 1$; (2) measures of variable importance unique to SIMCA; (3) how a sample is assigned to none, one or, if $Q > 1$, to several categories; and (4) characterization of prediction reliability. These matters are treated below in terms of a two-class case to simplify the discussion.

Between Class Measures

The PC models for the two classes are embodied by two trimmed loadings matrices, \mathbf{L}_1 and \mathbf{L}_2 . The class 1 samples can be fit to the class 2 model by projecting their transformed and preprocessed \mathbf{x} block, \mathbf{X}_1 , into the space defined by the class 2 loadings:

$$\hat{\mathbf{X}}_1 = \mathbf{X}_1 \mathbf{L}_2 \mathbf{L}_2^T \quad [6.3]$$

The residuals, \mathbf{E} , are defined as

$$\mathbf{E} = \hat{\mathbf{X}}_1 - \mathbf{X}_1 \quad [6.4]$$

where \mathbf{E} is a matrix containing a row vector \mathbf{e}_i for each class 1 sample. A between class residual for each sample in class 1 can be defined as:

$$s_{12} = \left(\frac{1}{(m - k_2)n_1} \sum_i^{n_1} \mathbf{e}_i \mathbf{e}_i^T \right)^{1/2} \quad [6.5]$$

where k_2 is the number of factors (*i.e.*, principal components) in the class 2 model and n_1 is the number of class 1 samples. The computation described by equation 6.5 is repeated for each pairwise combination of Q classes to yield a Q by Q matrix. Note that this matrix is not symmetrical, that is, $s_{12} \neq s_{21}$; fitting class 1 samples to the class 2 model is not equivalent to fitting class 2 samples to the class 1 model. For classes which are well-fit to themselves and well-separated from others, the diagonal matrix elements (which are the residuals of a class fit to itself) should be considerably smaller than the off-diagonal values.

Information about class separability can be presented in another way. For any pair of classes and the between class residual defined above, the between class distance is:

$$D_{12} = \left(\frac{\frac{s_{12}^2 + s_{21}^2}{2}}{\frac{s_{11}^2 + s_{22}^2}{2}} \right)^{1/2} - 1 = D_{21} \quad [6.6]$$

The computation described by [equation 6.6](#) can be repeated for each pairwise combination of Q classes to yield a Q by Q matrix. This matrix is symmetrical and the diagonal elements are all zero because $D_{11} = D_{22} = 0$. Large between class distances imply well-separated classes. A rule of thumb: classes are considered separable when the class distance is greater than 3⁸.

Measures of Variable Importance

It might be desirable to have a single metric of modeling power across all Q classes. Total modeling power is such a measure. It is analogous to the modeling power defined in [equation 5.41](#) and is derived from the variable variances defined in [equation 5.39](#) and [equation 5.40](#). For each class q, the variable residual variance is computed from the jth column of E:

$$\hat{s}_{jq}^2 = \frac{\hat{\mathbf{e}}_j^T \hat{\mathbf{e}}_j}{n_q - k_q - 1} \quad [6.7]$$

For each class, the total variance of that variable is also computed:

$$s_{0jq}^2 = \frac{1}{n_q - 1} \sum_i^{n_q} (x_{ij} - \bar{x}_j)^2 \quad [6.8]$$

The quantity in [equation 6.7](#) is corrected for the number of factors in each class model and summed over all classes:

$$s_{jT}^2 = \sum_q s_{jq}^2 \frac{m}{(m - k_q)} \quad [6.9]$$

The quantity in [equation 6.8](#) is also summed over all classes:

$$s_{0jT}^2 = \sum_q s_{0jq}^2 \quad [6.10]$$

The Total Modeling Power is then:

$$\text{TMP} = 1 - \frac{s_{jT}}{s_{0jT}} \quad [6.11]$$

Total Modeling Power is useful for determining which variables have little or no importance for any class in the training set. It typically ranges from 0 to 1 although it can become negative.

It may also be instructive to know which variables are best at discriminating between training set classes. For each variable, comparing the average residual variance of each class fit to all other classes and the residual variance of all classes fit to themselves provides an indication of how much a variable discriminates between “correct” and “incorrect” classification. The *Discrimination Power* is thus defined as:

$$DP_j = \frac{1}{Q-1} \frac{\sum_{q \neq r} \sum_{r=1}^Q (\hat{\mathbf{e}}_{jr})^T (\hat{\mathbf{e}}_{jr})}{\sum_{r=1}^Q (\hat{\mathbf{e}}_{jr})^T (\hat{\mathbf{e}}_{jr})} \quad [6.12]$$

The residual vector $\hat{\mathbf{e}}_{jr}$ denotes the j th column of the residual matrix after fitting training set samples in class r to the model for class q . The double sum in the numerator is evaluated for the cases in which q does not equal r .

Predicting the Class of an Unknown

An unknown (that is, a sample with no previously assigned category) can be fit to both the class 1 and class 2 models. When fit to class 1, it produces a residual vector:

$$\mathbf{e}_u = \hat{\mathbf{x}}_u - \mathbf{x}_u = \mathbf{x}_u \mathbf{L}_1 \mathbf{L}_1^T - \mathbf{x}_u \quad [6.13]$$

where the u subscript indicates the unknown. From the residual vector a residual variance can be calculated:

$$s_{u1}^2 = \frac{\mathbf{e}_u \mathbf{e}_u^T}{m-k} \quad [6.14]$$

The square root of this residual variance can be compared to a critical value calculated from

$$s_{crit1} = s_{01} (F_{crit})^{1/2} \quad [6.15]$$

where the critical F value is based on 1 and $n-k_1$ degrees of freedom and s_{01} is the square root of the class 1 variance as defined by:

$$s_{01}^2 = \frac{1}{n_1 - k_1} \sum_i^{n_1} s_i^2 \quad [6.16]$$

Note: If \mathbf{X} is mean-centered or autoscaled **and** the number of samples is less than or equal to the number of independent variables, all occurrences of the term $n-k$ become $n-k-1$.

If s_{u1} is less than s_{crit1} , the unknown is assigned to class 1.

The same process can be repeated for class 2: if s_{u2} is less than s_{crit2} , the unknown is assigned to class 2. If the unknown qualifies as a member of both classes, the class having the smallest sample residual is considered the best, and the other class is deemed next best. If the unknown exceeds both critical values, it is assigned to neither class.

Category prediction in SIMCA is parameterized in the same way as PCA; see “[Score Hyperboxes](#)” on page 5-30 for a discussion of the Standard Deviation Multiplier and Probability Threshold settings. Thus, when the SIMCA Standard Deviation Multiplier is changed (via Windows > Preferences > Prediction) from its default value of 0, the state-

ments in the previous paragraphs should be couched in terms of the square root of the augmented sample residual as defined by [equation 5.46](#).

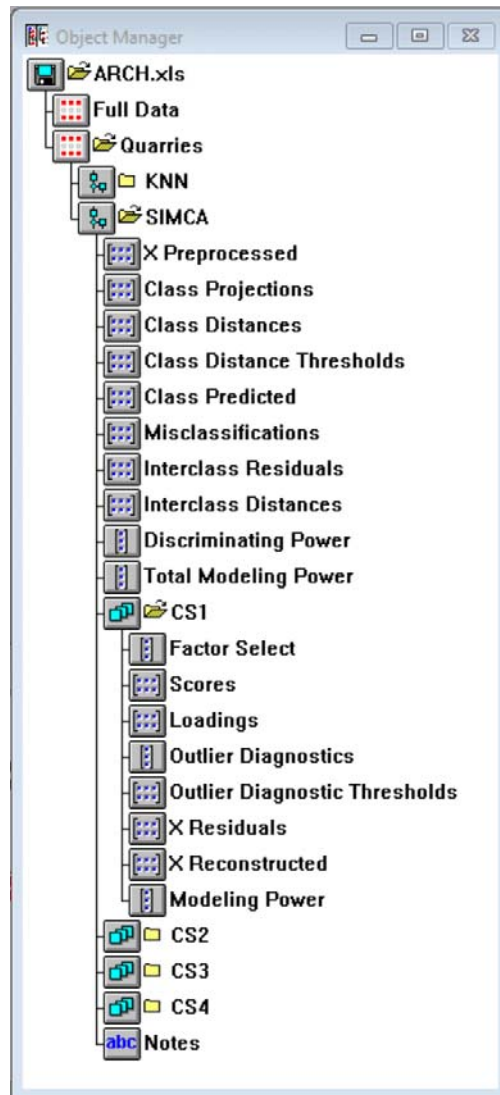
RUNNING SIMCA

Before running SIMCA, it is necessary to activate a class variable; for instructions and for the rules governing class variables, see [“Class Variables” on page 13-19](#). The algorithm options associated with SIMCA are described in [“SIMCA Options” on page 16-24](#). When the SIMCA algorithm executes, many computed objects are created and displayed. In addition, information necessary to make predictions is stored in memory as pieces of a classification model. A model can be used as soon as it has been created or it can be stored separately from the file containing the training set data and reloaded later to make predictions on future samples. A SIMCA model is more than just a set of trimmed loadings matrices for each class. It also contains information about which variables were excluded and what transforms/preprocessing options were chosen so that future samples are treated in the same way as the training set.

The objects computed during SIMCA can help you find sample outliers, choose the optimal number of factors for each class, and make decisions about excluding variables. Keep in mind that model building is an iterative process. You will seldom run a SIMCA algorithm just once and immediately start making predictions. Instead you will spend most of your time optimizing your model, that is, finding the “best” set of samples, variables and configuration parameters.

When SIMCA is run, objects particular to each training class (*i.e.*, the intraclass objects), are stored in folders separate from the interclass objects, as shown in [Figure 6.14](#). Because all of SIMCA’s intraclass objects are also created by PCA, the reader is again referred to discussions of [“Scores” on page 5-35](#), [“Loadings” on page 5-36](#), [“X Reconstructed” on page 5-45](#), [“X Residuals” on page 5-38](#), [“Outlier Diagnostics” on page 5-38](#) and [“Modeling Power” on page 5-39](#). The interclass objects described below will help you decide if the classes are sufficiently separated to produce reliable predictions once the class models have been optimized as described in [“Class Distances” on page 6-22](#).

Figure 6.14
SIMCA results in the
Object Manager



Interclass Residuals

The Interclass Residuals, described in “Between Class Measures” on page 6-16, are presented initially in a table view, as shown in Figure 6.15. Note that the column headings contain a suffix which reports the current number of factors set for that category’s model.

Figure 6.15
Interclass Residuals

7,1		1	2	3	4
		CS1@3	CS2@5	CS3@3	CS4@4
1	CS1	0.8638	7.6170	9.3459	4.0035
2	CS2	7.1731	0.2615	7.0955	3.6204
3	CS3	9.2496	5.2628	0.8531	4.4752
4	CS4	5.1238	8.1932	7.4618	0.6761

Interclass Distances

The Interclass Distances, described in “Between Class Measures” on page 6-16 and presented initially in a table view, is shown in Figure 6.16. Note that the column headings also contain a suffix which reports the number of factors for that category’s model.

Figure 6.16
Interclass Distances

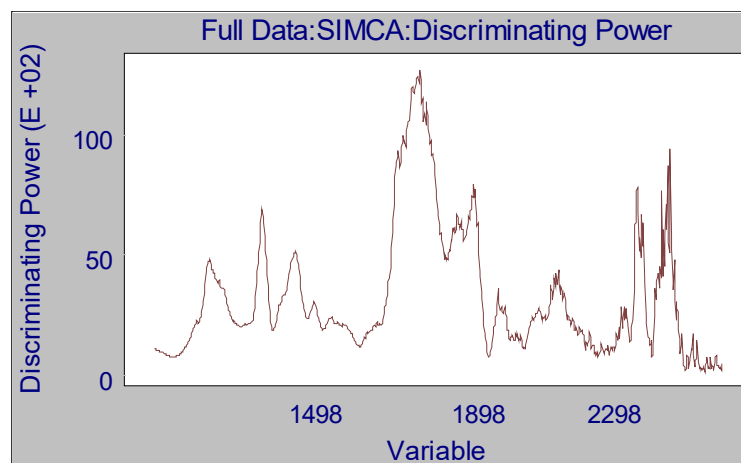
7,1					
		1	2	3	4
		CS1@3	CS2@5	CS3@3	CS4@4
1	CS1	0.0000	13.6654	11.1650	5.8629
2	CS2	13.6654	0.0000	8.9008	11.3565
3	CS3	11.1650	8.9008	0.0000	6.9934
4	CS4	5.8629	11.3565	6.9934	0.0000

As the distance between two class decreases, the likelihood of new samples classifying into both increases.

Discriminating Power

The quantity described by equation 6.12 is called the Discriminating Power; an example is shown below.

Figure 6.17
Discriminating Power



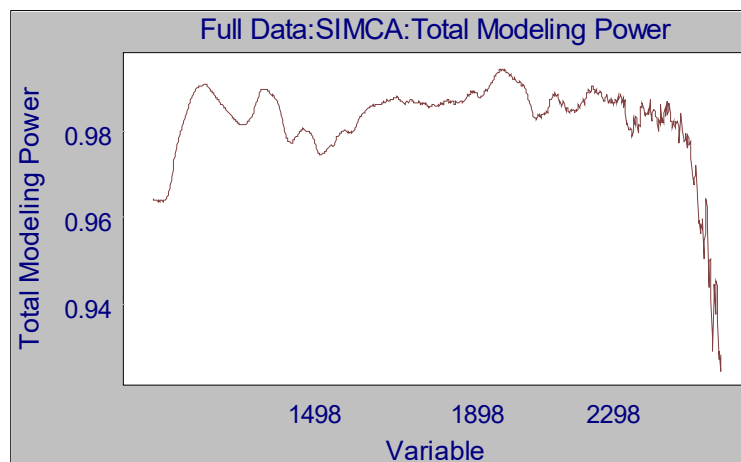
A value close to 0 indicates low discrimination ability in a variable, while a value much larger than 1 implies high discrimination power. In the above figure, the variables between 1700 and 1800 nm provide the most discriminating power.

Note: In contrast with the modeling power, it is not generally advised to exclude variables based solely on low Discrimination Power. Removing low discriminating variables has the effect of overly enhancing the separation between classes, at the expense of truly representative PC models.

Total Modeling Power

Total modeling power, shown next, is a measure of the importance of a variable to describe information from all training set classes.

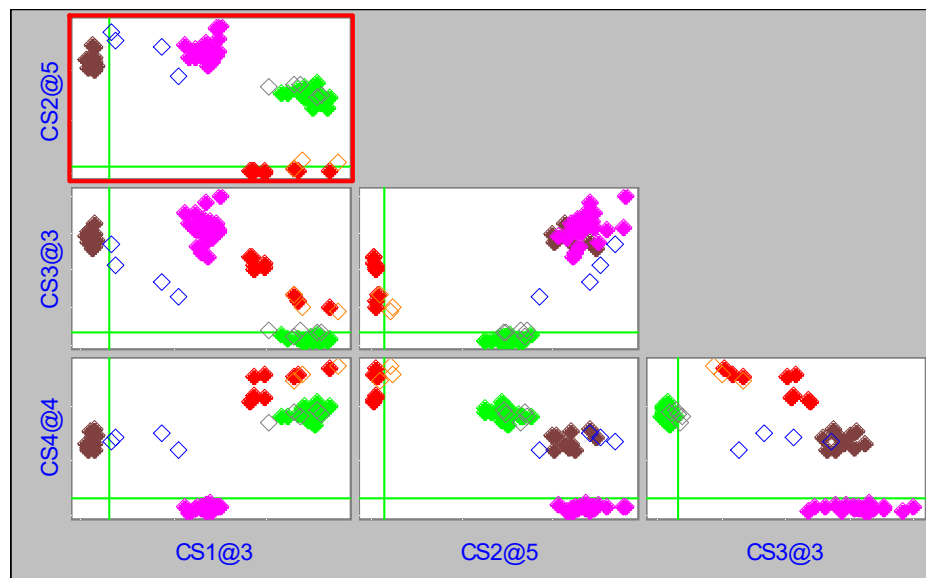
Figure 6.18
Total Modeling Power



Class Distances

During prediction (either the self-prediction that occurs when evaluating the training set or the prediction of unknowns), the (augmented) sample residual for each test sample fitted to each class is computed as described in “Predicting the Class of an Unknown” on page 6-18. The values, which are often called distances, are accumulated in the Class Distances object. By default, the information in this object is presented in a multiplot view where pairwise combinations of classes form the subplot axes. Such a multiplot is shown below.

Figure 6.19
Multiplot of Class Distances

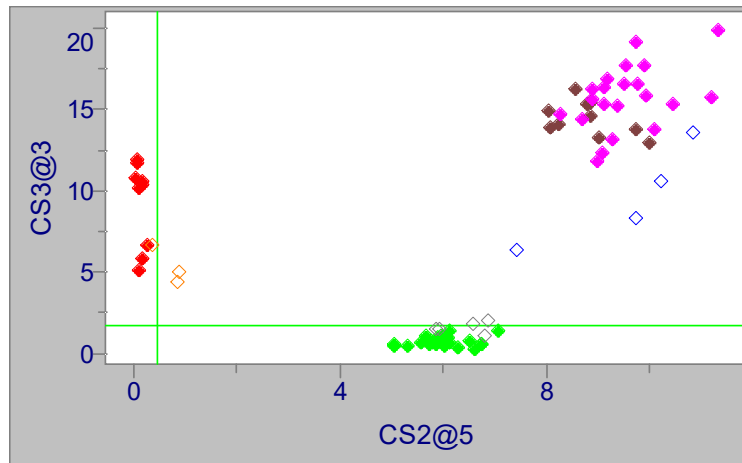


A single subplot when zoomed results in a display like that in the following figure. The two threshold lines are the s_{crit} values for each training set. Any sample can be visually classified by observing its position in the appropriate class distance subplot.

The threshold lines divide the plot into four quadrants. A sample in the NW quadrant is a member only of the x axis class; its distance to that class is small enough for it to be considered a member of the class. A sample falling in the SE quadrant is a member only of the y axis class. A sample in the SW quadrant could belong to either category and one

in the NE quadrant belongs to neither. These plots can be thought of as decision diagrams, as described by Coomans⁹. They present classification information visually and also draw attention to borderline cases, samples lying close to one or both thresholds.

Figure 6.20
Class Distances subplot



Class Predicted

Because SIMCA can classify samples in one of three ways, the predicted class of a sample may be more complicated than the TestVote single category assignment produced by KNN; it may contain more than one class assignment. The object shown below comes from a training set containing several categories.

Figure 6.21
Class Predicted showing a sample fit to two classes

Full Data:SIMCA Predict:Class Predicted		5.000000		
		1	2	3
		Best	NextBest1	
9	R01538	2	0	
10	cs2outlier	0	0	
11	T239	2	0	
12	T310	2	0	
13	T328	2	0	
14	T335	2	0	
15	R82710	5	6	
16	R82728	5	6	
17	R83403	5	6	
18	R04482	5	0	
19	R04827	5	6	
20	1304	5	0	

When a sample qualifies as a member of one class and as a nonmember of all other classes, a straightforward assignment is possible. For example, the 18th sample in the table is

assigned unambiguously to category 5 while the 9th fits only category 2. We can tell that the assignments are unambiguous because the NextBest category for these samples is 0.

Note: A predicted class of 0 signifies no matching training set category. However, you cannot assign training set samples to class 0 and then run SIMCA with this class. See, also, “Class Variables” on page 13-19.

A non-zero value in the NextBest column indicates that some samples qualify for membership in more than one class. The 17th sample in the table is assigned to two categories although it fits category 5 better than category 6.

Unlike KNN, it is also possible that with SIMCA an outcome is that a sample does not match any category. For example, sample 10 in the above table fit none of the classes in the training class.

Misclassifications

Many of the previously mentioned objects address data suitability. These objects help us determine if the samples in a category are homogeneous and representative and if the variables are useful and descriptive. However, the classification bottom line is how well did we do? Particularly during training, we would like to evaluate the SIMCA model’s self-consistency. Pirouette provides a Misclassification object (much like the KNN object of the same name discussed in “Misclassifications and Total Misses” on page 6-8) to summarize classification success by category. From the Best column of the Class Predicted object, a cross-tabulation is constructed, such as that shown in the following figure.

Figure 6.22
SIMCA
Misclassifications

	Pred2@4	Pred3@4	Pred4@4	Pred5@4	Pred6@4	No match
Actual2	12	0	0	0	0	0
Actual3	0	16	0	0	0	0
Actual4	0	0	12	0	0	0
Actual5	0	0	0	16	0	0
Actual6	0	0	0	0	11	0

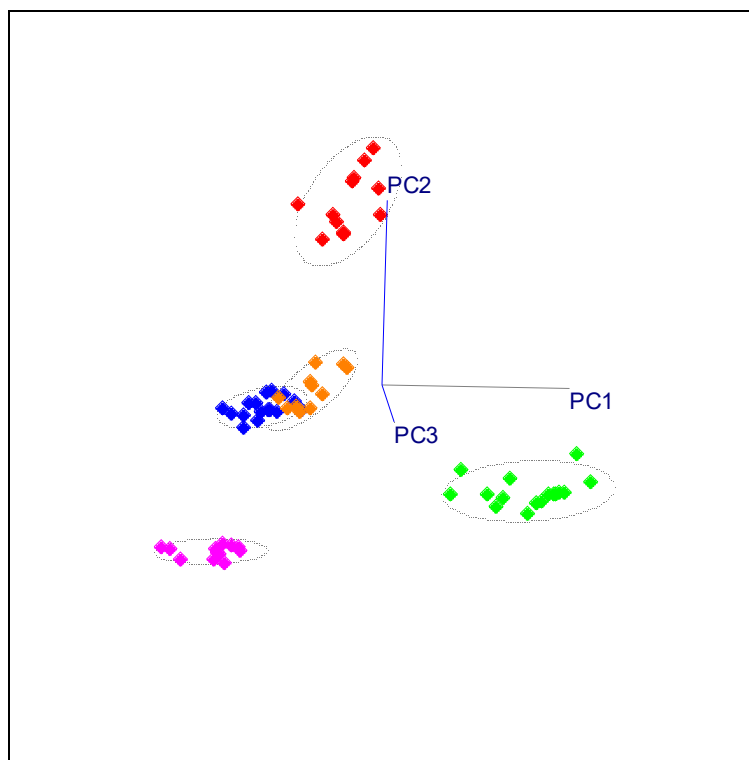
If samples in the training set categories are reasonably homogeneous, no misclassifications should occur (as in the table above). In theory, outliers in the training set may not properly self-predict. Either these samples would be classified into the wrong category and a non-zero value would occur off the table’s diagonal, or they would not match any category, in which case a non-zero value would be placed in the last column, titled *No match*. In practice, however, most training set samples properly self-predict for one obvious reason: the sample’s data was used to model the category.

Class Projections

The Class Projections object provides a visual evaluation of the degree of class separation. To create this object, a 3 factor PCA is performed on the entire training set during the SIMCA processing. The coordinates of a bounding ellipse (based on the standard deviations of the scores in each PC direction) for each category are projected into this 3 factor PCA space; they form a confidence interval for the distribution of the category. The figure below shows the score points for four categories and the corresponding confidence intervals displayed as ellipses. Rotation of the Class Projections plot can reveal category

overlap as well as training set samples lying beyond the confidence boundary of the corresponding class model.

Figure 6.23
A Class Projections
plot from a SIMCA
analysis



OPTIMIZING THE MODEL

Model optimization in SIMCA differs markedly from KNN in that each training class in SIMCA is considered independently. Determining the optimal number of factors in each class and finding outliers is accomplished in SIMCA in a manner analogous to that described for PCA (save for the fact that cross-validation is not currently implemented in Pirouette's SIMCA algorithm).

Admittedly, quite a few objects are calculated during SIMCA model creation and examining them all can be tedious when the number of classes is large. Each can inform your choice of the optimal model parameters. Viewing the scores in rotatable 3D will help you decide if classes are suitably homogeneous. Make sure to scan the Outlier Diagnostics plot for unusual samples.

Variable selection may be necessary to improve class separation or to reduce the complexity of individual classes. For this, examine the modeling power and the discrimination power, as well as the loadings for each category. Even if these parameters appear satisfactory, you should also verify that there is class separability by looking at the interclass residuals and distances. If the distances between classes are small, you may find it difficult to obtain reliable classifications.

Remember the iterative nature of model building. For each class, as you change the number of factors via a line plot of the Factor Select object, you must track the impact on objects linked to the number of factors: Outlier Diagnostics and Modeling Power for that class and all interclass objects. If you change transforms or preprocessing options or ex-

clude samples or variables, optimization must be repeated after re-executing the SIMCA algorithm.

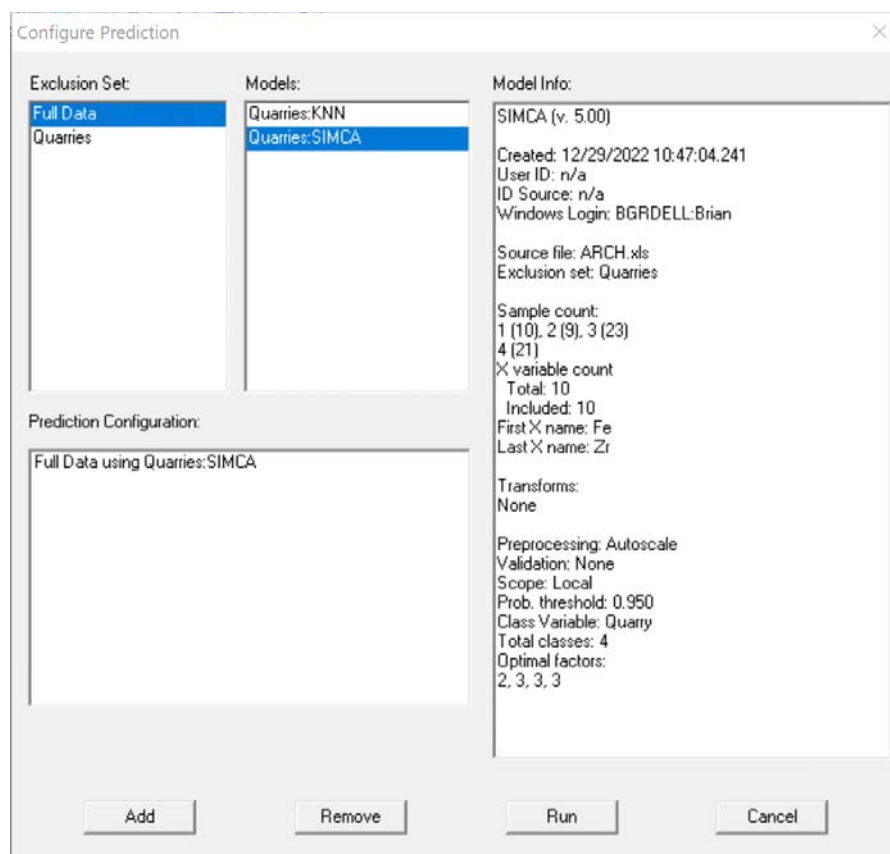
In cases where only two of many classes are insufficiently distinguished, you may want to create an exclusion set containing only those two classes and treat them separately, eliminating variables where no difference exists. Then you can use one SIMCA model to make the easier distinctions and a second model to focus only on the difficult ones. This process is sometimes called hierarchical classification and is an approach for which the automation software InStep is well-suited.

MAKING A SIMCA PREDICTION

Running SIMCA triggers the creation of a model. You can confirm this by going to Process/Predict after running either algorithm and noting the entry under Model. Making predictions requires a model and a target, that is, a data set with an x block containing one or more samples whose classes will be predicted. This data set's x block must have the same number of independent variables as the data set from which the model was created and cannot contain any excluded independent variables. The prediction target may or may not contain dependent and class variables.

The following figure shows the Configure Prediction dialog box. To get model information, highlight its entry as illustrated in the figure. To configure a prediction, highlight a model and an exclusion set and click on Add. You can configure more than one prediction at a time by highlighting a different model name or exclusion set then clicking Add. Predictions are made when you click Run. Some prediction objects summarize results for all classes; others are stored in a folder for each training set category.

Figure 6.24
Configure Prediction
dialog box

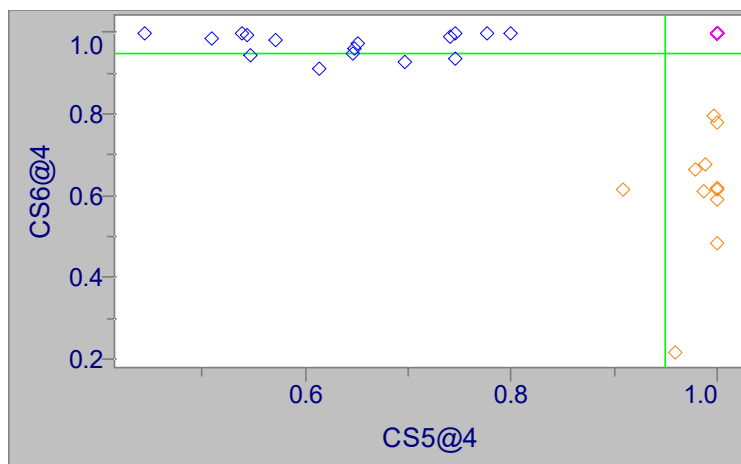


For each training set class, Scores, X Reconstructed, and X Residuals are computed for each test sample and stored in a separate folder. All are discussed in “Making a PCA Prediction” on page 5-43.

Class Probabilities

During prediction, a probability can be computed from the (augmented) sample residual for each test sample fitted to each class as described in “Outlier Diagnostics” on page 5-38. The values are accumulated in the Class Probabilities object. By default, the information in this object is presented in a multiplot view (like the Class Distances object) where pairwise combinations of classes form the subplot axes. A zoomed subplot is shown in the following figure.

Figure 6.25
Class Probabilities
subplot



These Class Probabilities characterize a sample’s quality of fit to the various SIMCA classes much like the Class Distances discussed on page 6-22: samples laying in the NW quadrant are members of the x-axis category, etc.

Misclassifications

The prediction Misclassifications object differs from the modeling analog in one respect. The table contains an additional line--Unmodeled--which shows the predicted category for those samples not assigned an *a priori* category value or assigned to a category not present in the model. An example result follows.

Note: *Misclassifications are computed only if there is a Class variable in the prediction set which has the same name as the Active Class used during the SIMCA modeling phase. The name match is case sensitive.*

Figure 6.26
SIMCA
Misclassifications
during prediction

	1	2	3	4	5
	Pred1	Pred2	Pred3	Pred4	No match
Actual1	10.0000	0.0000	0.0000	0.0000	0.0000
Actual2	0.0000	8.0000	0.0000	0.0000	1.0000
Actual3	0.0000	0.0000	21.0000	0.0000	2.0000
Actual4	0.0000	0.0000	0.0000	21.0000	0.0000
Unmodeled	1.0000	3.0000	2.0000	0.0000	6.0000

In the example above three samples did not achieve the expected classification (those in the No match column), although the majority of samples were properly classified. Borrowing the terminology of clinical analysis, the classification outcomes can be further summarized in four ways:

- True positive (tp) - a sample known to be a member of category A classifies *correctly* into category A
- False negative (fn) - a sample known to be a member of category A classifies *incorrectly* into a different category or into no model category
- False positive (fp) - a sample known not to be a member of category A classifies *incorrectly* into category A
- True negative (tn) - a sample known not to be a member of category A classifies *correctly* into another category or into no model category

The above table contains 8 true positive samples of category 2, one false negative and 3 false positives. The remaining 63 samples are true negatives of category 2.

For purposes of comparing results between different classification models, these outcomes can be characterized by their sensitivity and selectivity. For a given category:

$$\text{Sensitivity} = \frac{\text{tp}}{\text{tp} + \text{fn}} \quad [6.17]$$

Thus, sensitivity can be calculated from the values found in the misclassification matrix as the ratio of the number of samples in the cell where the categories for predicted and actual columns are the same divided by the number of samples in the row for that category.

$$\text{Selectivity} = \frac{\text{tn}}{\text{tn} + \text{fp}} \quad [6.18]$$

Similarly, selectivity can be found as the ratio of the total number of samples minus the number of samples in the category minus the number of unmodeled samples predicted into the category, divided by the total number of samples minus the samples in the category.

We can extend this evaluation to the entire prediction set. The sensitivity is the ratio of the sum of the diagonal values in the Actual vs. Predicted portion of the table to the sum of all of the Actual rows. In the table above, the sensitivity would be 60/63 or 95%. The selectivity simplifies to the ratio of the Unmodeled-No match value to the sum of the Unmodeled row: 6/12 or 50%.

The user must determine whether the consequence of a false negative (lower sensitivity) outweighs the consequence of a false positive (lower selectivity).

Calibration Transfer

Classification models, when saved to a file, are portable and can be shared with other Pirouette users. However, differences among instruments may be great enough to make predictions using shared models unreliable. Transfer of calibration approaches may allow such models to be used with little or no loss of reliability. For more background on this topic, see “[Calibration Transfer](#)” in [Chapter 4](#).

To transfer a calibration during classification prediction, you must use a model derived from an algorithm configured to Enable Calibration Transfer. The check box for this option appears in [Figure 6.2, on page 6-5](#). Along with the profiles (that is, the x block) to be adjusted, you must also supply two class variables and choose the calibration transfer type. The contents of and constraints on these two variables are described below.

REQUIRED CLASS VARIABLES

The Adjust Mask class variable determines which prediction samples are candidate transfer samples; it must contain only 1s and 0s and must contain at least one 1. Samples flagged with a 0 are excluded from calibration transfer calculations; samples flagged with a 1 may be involved, depending on other conditions. The name of the Adjust Mask variable is specified in Prediction Preferences dialog (see “[Prediction](#)” on [page 10-19](#)).

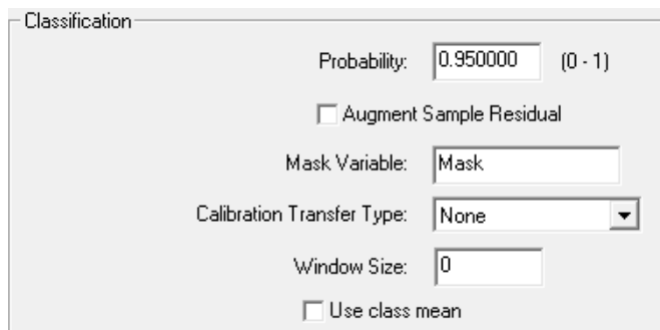
For a calibration to be transferred, it is mandatory that the prediction set include a class variable with exactly the same name as in the model. For each candidate transfer sample, the value of this class variable is examined and compared to the class values in the model of the training set. If a value is found which is not in the model, the calibration cannot be transferred and the prediction aborts. Neglecting to supply transfer samples from all model categories compromises the efficacy of the transfer as does supplying too few samples per category. The number of transfer samples per category actually used for the transfer is determined by the category with the smallest number of transfer samples, whether it be in the model or in the prediction set.

Note: *If it is not possible to supply transfer samples for every model category, the Additive and Multiplicative adjust types are usually preferable to Direct and Piecewise.*

CALIBRATION TRANSFER OPTIONS

The last three items in the Classification group of the Prediction Preferences dialog (see below) apply to transfer of calibration. There you specify the name of the Adjust Mask Variable, the Adjust Type and Window Size, the latter applicable only if Piecewise is chosen as the transfer type. For background on the different types of transfer, see “[Calibration Transfer](#)” on [page 4-33](#).

Figure 6.27
Classification
Prediction
Parameters



Classification

Probability: 0.950000 (0 - 1)

Augment Sample Residual

Mask Variable: Mask

Calibration Transfer Type: None

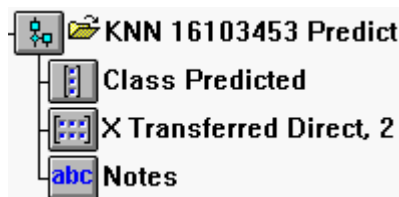
Window Size: 0

Use class mean

X TRANSFERRED

When a calibration is transferred, the results include an additional object, which is named X Transferred, with the transfer type parameter and number of samples per category appended. For example, the figure below shows that Direct transfer was applied with 2 transfer samples per category. This object contains the x block of the prediction samples after adjustment. You should always compare this object to the original transformed prediction profiles. Similarly, it is wise to compare the predicted categories with and without adjustment.

Figure 6.28
X Transferred object



References

1. Anderson, T.W.; *An Introduction to Multivariate Statistical Analysis* (Wiley: New York, 1958).
2. Nilsson, N.J.; *Learning Machines* (McGraw-Hill: New York, 1965).
3. Lachenbruch, P.A.; *Discriminant Analysis* (Haffner Press: New York, 1975).
4. Kowalski, B.R. and Bender, C.F.; "Pattern Recognition. A Powerful Approach to Interpreting Chemical Data," *J. Am. Chem. Soc.*, (1972) 94: 5632.
5. Beebe, K.R.; Pell, R.J.; and Seasholtz, M.B.; *Chemometrics: A Practical Guide*, (John Wiley & Sons: New York, 1998).
Derde, M.P. and Massart, D.L. "Supervised pattern recognition: the ideal method?" *Anal. Chim. Acta*, (1986) 191:1-16.
6. Wold, S.; "Pattern Recognition by Means of Disjoint Principal Components Models," *Pattern Recognition*, (1976) 8: 127-139.

7. Forina, M and Lanteri, S.; "Data Analysis in Food Chemistry." in B.R. Kowalski, Ed., *Chemometrics. Mathematics and Statistics in Chemistry* (D. Reidel Publishing Company, 1984), 305-349.
8. Kvalheim, O.M. and Karstang, T.V.; "SIMCA - Classification by Means of Disjoint Cross Validated Principal Components Models." in R.G. Brereton, Ed., *Multivariate pattern recognition in chemometrics, illustrated by case studies* (Elsevier: Amsterdam, 1992), p. 237.
9. Coomans, D. and Broeckaert, I.; *Potential Pattern Recognition in Chemical and Medical Decision Making*, Research Studies Press LTD (John Wiley: Letchworth, England, 1986), p. 215.

Reading List

1. Derde, M.P.; Coomans, D. and Massart, D.L.; "Effect of Scaling on Class Modeling With the SIMCA Method." *Anal. Chim. Acta*, (1982) 141: 187-192.
2. Derde, M.P. and Massart, D.L.; "Comparison of the Performance of the Class Modelling Techniques UNEQ, SIMCA, and PRIMA," *Chemometrics and Intelligent Laboratory Systems*, (1988) 4:65-93.
3. Sharaf, M.A.; Illman, D.L.; and Kowalski, B.R.; *Chemometrics* (Wiley: New York, 1986).
4. Shah, N.K. and Gemperline, P.J.; "Combination of the Mahalanobis Distance and Residual Variance Pattern Recognition Techniques for Classification of Near-Infrared Reflectance Spectra," *Anal. Chem.*, (1990) 62:465-470.
5. Sjostrom, M. and Kowalski, B.R.; "A Comparison of Five Pattern Recognition Methods Based on the Classification Results from Six Real Data Bases," *Anal. Chim. Acta*, (1979) 112:11-30.
6. Wold, S.; Albano, C.; Dunn III, W.J.; Edlund, U.; Esbensen, K.; Geladi, P.; Hellberg, S.; Johannsson, E.; Lindberg, W.; and Sjostrom, M. "Multivariate Data Analysis in Chemistry." in Kowalski, B.R., Ed., *Proceedings NATO Adv. Study Inst. on Chemometrics Cosenza*, Italy, Sept. 1983 (Reidel Publ. Co: Dordrecht, Holland, 1984), pp. 17-95.
7. Wold, S. and Sjostrom, M.; "SIMCA: A Method for Analyzing Chemical Data in Terms of Similarity and Analogy." in B.R. Kowalski, Ed., *Chemometrics: Theory and Application*, ACS Symposium Series 52 (American Chemical Society: Washington, DC, 1977), 243-282.

Regression Methods

Contents

Factor Based Regression	7-2
PLS for Classification	7-38
Classical Least Squares	7-44
Calibration Transfer	7-56
Locally Weighted Regression	7-58
References	7-59

The ultimate purpose of most multivariate analyses is to develop a model to predict a property of interest. That property may be categorical or continuous. Developing models for category prediction is the subject of Chapter 6. Continuous properties are modeled and predicted by regression methods, the subject of this chapter.

Regression establishes a functional relationship between some quantitative sample property, the dependent variable, and one or more independent variables. In analytical chemistry, the independent variables are often chromatographic or spectroscopic measurements. It would be advantageous to substitute these measurements for bulk property or concentration determinations which are problematic due to either high cost or the lack of a specific sensor. Examples of a bulk property are the flash point of a fuel and the strength of a polymer. Indirectly determined concentrations include moisture, fat, fiber and protein in food and oxygenates in gasoline. Compared to univariate regression, multivariate methods offer improved precision, more sophisticated outlier detection and in the case of factor based algorithms, the possibility of compensating for interferences.

Many users of regression are content to create descriptive models. They apply the techniques discussed in this chapter but stop when a model has been created. By describing the correlations among the variables, these models can point out phenomena which give rise to the structure in the data. In these scenarios, there is no future; the data in hand define the problem completely. If more data become available later, they are either added to the existing pool of data or are modeled separately.

We, however, have a more ambitious purpose: predictive models. In this case, modeling dependent variables in terms of independent variables is only the first step. Later, the model is combined with independent variable measurements on new samples (sometimes called unknowns) to predict their dependent variable properties. The data used to create a predictive model are called the training set and the model-building phase is often referred to as calibration. The term *calibration model* emphasizes that prediction, not description, is the ultimate objective.

After a model is created but before it is used, it should be validated. Validation, which establishes the reliability of a predictive model, entails making predictions on samples with already known dependent variable settings. A model may describe the training set very well but yield poor predictions when applied to future samples. Thus, validation is as important as the model-building process itself because it tells us how well a model should perform.

Suppose we wish to predict the concentrations of components A and B in unknowns also containing components C, D and E but we wish to avoid quantifying C, D and E. The two factor based algorithms discussed below, PLS and PCR, can address this not uncommon analytical scenario. At first glance this claim may seem fantastic to chemists familiar with the concept of an interference. However, because factor based methods characterize variance, they produce linear combinations of independent variables which account for variation associated with unquantified components even without explicit information about dependent variable settings for those components. Of course, they work only if all components in the unknowns are present in the training set in amounts spanning the ranges encountered in the unknowns. To elaborate, the training set must include samples containing compounds A, B, C, D and E; the amounts of A and B must be known; and the concentrations over which all components vary must span the range encountered in future unknowns. These methods, which are described as both implicit and inverse, are said to produce soft models.

The obvious power of factor based methods is mitigated by two demands which immediately catch the attention of those new to them. First, a relatively large number of training set samples is required to adequately discern the variance patterns. Second, the user must decide how many factors are necessary to model not only the properties to be predicted but also the interferences. Pirouette's two factor based algorithms are discussed in [“Factor Based Regression” on page 7-2](#).

Some users, particularly spectroscopists, may live in (or occasionally visit) a simpler world: applications where all sources of variation are known and quantified. In this case, a so-called classical method can be employed which is said to produce hard models. See [“Classical Least Squares” on page 7-44](#) for a discussion of Pirouette's implementation of CLS.

Factor Based Regression

Pirouette implements two popular multivariate regression methods, both of which are factor-based: Principal Component Regression (PCR) and Partial Least Squares (PLS) regression. While both produce a lower dimension representation of the independent variable block, they differ in how this representation is computed. The ramifications of these differences are not always apparent so a user is often left wondering if one is better. For many data sets, neither method significantly outperforms the other in producing reliable models, which is after all the bottom line. In fact, the results are often indistinguishable at first glance except that PLS may produce a model with one less factor than PCR. Moreover, in Pirouette, PLS executes faster than PCR. These two facts lead many to prefer PLS over PCR. This said, it should also be noted that PCR is better understood from a statistical point of view.

Pirouette's PLS algorithm is often referred to as PLS1; it does not perform a PLS2-type of decomposition in which multiple dependent variables are processed as a unit. For details on the theory and application of PLS and PCR, refer to Martens and Næs¹.

MATHEMATICAL BACKGROUND

Before describing how to apply PLS and PCR, we supply a brief theoretical background for those new to factor-based regression, The references at the end of this chapter address the topics below in much greater depth. As in Chapter 5 and Chapter 6, we consider the case where m measurements are made on n samples. For each sample, the m independent variables are arranged in a row vector. For our purposes, these row vectors are assembled into a matrix containing n rows and m columns called the X (or independent variable) block. The Y (or dependent variable) block contains at least one column vector having n elements. Several dependent variables can be associated with an X block but each is processed separately. In the following discussion, i is a sample index, j is a dependent variable index and k is a factor index. The maximum number of latent (or abstract) factors, g , is equal to the minimum of n and m .

Multilinear Regression

To predict some value y from a suite of other measurements x_j (where $j = 1, 2, \dots, m$), we must first establish a relationship between the two sets of measurements. If we assume that y is linearly related to x and write:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m + f \quad [7.1]$$

then the beta terms (which are called regression coefficients) specify the relationship we seek, and f contains the error in describing this relationship. For a set of n samples ($i = 1, 2, \dots, n$):

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_m x_{im} + f_i \quad [7.2]$$

In matrix format (with mean-centering to remove the first beta term), this becomes:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{f} \quad [7.3]$$

The error vector, \mathbf{f} , is included because it is unlikely that y can be expressed exactly in terms of the X block; f_i is the y residual for the i th sample. The determination of the vector of regression coefficients allows future y values to be predicted from future X block measurements; thus, finding the beta vector is described as creating a regression model. The regression coefficients satisfy a least squares criterion: they minimize of the error sum of squares defined as:

$$\mathbf{f}^T \mathbf{f} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \quad [7.4]$$

where the T superscript indicates the transpose of the matrix. Thus,

$$\mathbf{y} - \mathbf{X}\boldsymbol{\beta} = 0 \quad [7.5]$$

To meet this condition:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad [7.6]$$

To make a prediction of y from a new \mathbf{x} , we substitute [equation 7.6](#) into [equation 7.3](#):

$$y_{\text{new}} = \mathbf{x}_{\text{new}} \boldsymbol{\beta} = \mathbf{x}_{\text{new}} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad [7.7]$$

When there are more samples than variables, this approach is referred to as Multiple Linear Regression (MLR) or Inverse Least Squares (ILS). Note that it requires inversion of

an m by m matrix. If the columns of \mathbf{X} are linearly dependent or if they are highly correlated, the matrix is singular or almost singular. What this means practically is that the computed regression coefficients are relatively imprecise. As you can imagine, predictions based on a model with poorly-determined regression coefficients are likewise imprecise. This issue is avoided by using a factor based method (see below) in which the variables are by definition uncorrelated.

Many analytical instruments produce highly correlated measurements. For example, in many spectroscopic techniques, signal changes at one wavelength are often accompanied by similar changes at other wavelengths. Correlated variables are good from the standpoint of redundancy; acting in some sense like replicates. However, violation of the MLR assumption that each x variable is independent can cause the algorithm to fail when applied to data sets containing highly correlated variables. If MLR is the only technique under consideration, the user must find and eliminate correlated variables. Thus, the emphasis in MLR is often on variable selection, deciding which subset of the original independent variables data set to retain and which to discard.

Factor based methods are an alternative to MLR. As discussed in “[Principal Component Analysis](#)” in [Chapter 5](#), they find linear combinations of the independent variables which account for variation in the data set. These linear combinations, which are sometimes called loadings or latent variables, are the factors which give the approach its name. They can be thought of as new variables which have the desirable property of being uncorrelated. There are many factor based techniques which differ only in how they define a factor, that is, how the linear combinations of the original variables are found. An advantage to these techniques is that no data have to be discarded. Disadvantages include difficulty in interpreting the factors and the need to decide how many factors to compute.

Principal Component Regression

In PCR the independent variable block (*i.e.*, the \mathbf{X} block) is first decomposed as in PCA. However, in Pirouette PCR employs the Singular Value Decomposition (SVD)² rather than the NIPALS algorithm (“[NIPALS](#)” on [page 5-28](#)). In the SVD, the matrix \mathbf{X} is decomposed into three matrices:

$$\mathbf{X} = \mathbf{USV}^T \quad [7.8]$$

The \mathbf{U} matrix holds the eigenvectors of the row space, the \mathbf{V} matrix holds eigenvectors of the column space, and \mathbf{S} is a diagonal matrix whose diagonal elements are the singular values. A singular value is the square root of the corresponding eigenvalue of the product of $\mathbf{X}^T\mathbf{X}$. There are as many singular values as samples or independent variables, whichever is smaller. One advantage of such a decomposition is that a potentially troublesome inversion is avoided. Note that SVD results are interconvertible with PCA results:

$$\mathbf{L} \equiv \mathbf{V} \quad [7.9]$$

$$\mathbf{T} \equiv \mathbf{US} \quad [7.10]$$

Replacing \mathbf{X} by its decomposition, we can then proceed with the regression. First, we substitute [equation 7.8](#) into [equation 7.3](#):

$$\mathbf{y} = (\mathbf{USV}^T)\boldsymbol{\beta} + \mathbf{f} \quad [7.11]$$

The solution then becomes:

$$\beta = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{y} \quad [7.12]$$

where this β term is the regression vector. Predicting y from a new \mathbf{x} follows from:

$$y_{\text{new}} = \mathbf{x}_{\text{new}}\beta = \mathbf{x}_{\text{new}}\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{y} \quad [7.13]$$

A notable aspect of PCR is that the SVD decomposition of \mathbf{X} depends only on \mathbf{X} . The \mathbf{Y} block has no influence on \mathbf{U} , \mathbf{S} or \mathbf{V} , only on β .

Partial Least Squares Regression

Partial Least Squares (PLS) regression shares many PCR/PCA characteristics and vocabulary. PLS (also known as Projection to Latent Structures) finds factors analogous to PCA's principal components. However, because these factors contain information about \mathbf{X} and \mathbf{Y} block correlations, they often yield more parsimonious and, in some situations, more reliable models than PCR.

The original descriptions of PLS were based on the NIPALS algorithm³. Later, PLS was shown to be equivalent to a matrix bidiagonalization⁴, which is how Pirouette implements PLS. The interested reader is referred to the Manne paper for details. The bidiagonalization matrices are analogous to those derived from the SVD. To minimize confusion which might result from PCR/PLS analogs sharing symbols, an underscore will denote PLS matrices:

$$\mathbf{X} = \underline{\mathbf{U}}\mathbf{R}\underline{\mathbf{V}}^T \quad [7.14]$$

$$\underline{\mathbf{L}} \equiv \underline{\mathbf{V}} \quad [7.15]$$

$$\underline{\mathbf{T}} \equiv \underline{\mathbf{U}}\mathbf{R} \quad [7.16]$$

$\underline{\mathbf{U}}$ and $\underline{\mathbf{V}}$ are not identical to \mathbf{U} and \mathbf{V} , which means that PLS scores/loadings are different from PCR scores/loadings, although they are often quite similar. One detail worth noting is the orthogonality (or the lack thereof) of the analogs: both loadings are orthogonal as is $\underline{\mathbf{T}}$ but the NIPALS scores from PLS are not. \mathbf{R} is not a diagonal matrix as \mathbf{S} is in PCR; rather, it is right bidiagonal, *i.e.*, the elements on the first diagonal above the main diagonal may be nonzero. The diagonal elements of the \mathbf{R} matrix are not equivalent to SVD singular values. A pseudo-eigenvalue vector can, however, be calculated from the PLS scores:

$$\underline{\lambda} = \text{diag}(\underline{\mathbf{T}}^T\underline{\mathbf{T}}) \quad [7.17]$$

Note: The term **pseudo-eigenvalue** is employed to avoid confusion with the term **eigenvalue**, which has a specific mathematical meaning. PLS pseudo-eigenvalues are similar to the actual eigenvalues determined in PCR in that they quantify the amount of variation accounted for by a factor. In the case of PCR, variation means \mathbf{X} block variation; in PLS, variation also includes \mathbf{Y} block effects. Having made this distinction, we omit the pseudo modifier when referring to this PLS result below.

The PLS regression step is carried out exactly as in PCR, except that [equation 7.14](#) is substituted into [equation 7.3](#):

$$\beta = \underline{\mathbf{V}}\mathbf{R}^{-1}\underline{\mathbf{U}}^T\mathbf{y} \quad [7.18]$$

Predicting y from a new \mathbf{x} follows from:

$$y_{\text{new}} = \mathbf{x}_{\text{new}}\boldsymbol{\beta} = \mathbf{x}_{\text{new}}\mathbf{V}\mathbf{R}^{-1}\mathbf{U}^T\mathbf{y} \quad [7.19]$$

NIPALS Scores and Loadings

The vigilant user will note that the scores and loadings computed by the bidiagonalization algorithm differ slightly from those produced by the NIPALS algorithm. In Pirouette, the PLS NIPALS results are called NIPALS Scores and NIPALS Loadings. They are provided only for comparison purposes. A NIPALS user who wants to check results against Pirouette should examine the NIPALS analogs. NIPALS was developed years before bidiagonalization, so when “PLS scores” and “PLS loadings” are mentioned in the literature, the NIPALS analogs are usually meant. The bidiagonalization loadings are usually called weight loadings by NIPALS users. Of course, when PLS regression is performed, a consistent set of scores and loadings must be chosen. NIPALS scores are orthogonal while NIPALS loadings are not, just the converse of bidiagonalization. More importantly, the computation of the \mathbf{X} residuals will also differ with the two approaches (see note on [page 7-11](#)).

Note: *There is an alternative formulation of the PLS algorithm given in the Martens book¹ (on page 123) which they refer to as the non-orthogonalized PLSR algorithm. This alternative produces scores and loadings identical to those in the bidiagonalization approach.*

Trimmed Matrices

If \mathbf{X} is fully decomposed into its g factors, the regression vectors of [equation 7.12](#) and [equation 7.18](#) are identical to that of [equation 7.6](#). However, if only the first k columns of \mathbf{U} (or \mathbf{U}), \mathbf{S} (or \mathbf{R}) and \mathbf{V} (or \mathbf{V}) are kept, a k factor approximation of \mathbf{X} is produced. This process, often called reducing the dimensionality of the data set, is discussed in “[Modeling with PCA](#)” on [page 5-18](#) in terms of trimmed scores and loadings matrices, \mathbf{T}_k and \mathbf{L}_k , respectively. (Refer to [equation 7.9](#) and [equation 7.10](#) for the relationship between PCR scores/loadings matrices and the SVD decomposition; refer to [equation 7.15](#) and [equation 7.16](#) for the relationship between PLS scores/loadings matrices and the bidiagonalization decomposition.) Replacing the original \mathbf{X} matrix with a lower dimension approximation is the central idea of factor based regression. By taking this step, a significant advantage is achieved: the resulting regression coefficients no longer suffer from the large relative uncertainties sometimes associated with MLR coefficients. Of course, there is a price: the user must set k which then determines the regression coefficients and all of its derived quantities.

Estimating the Optimal Number of Factors

Dealing with unvalidated regression models is presented in “[Estimating the Number of Factors in Unvalidated Models](#)” in [Chapter 5](#). However, a different approach is taken when cross-validation is specified.

Validation-Based Criteria

We typically create models in order to make predictions on future data. If we retain an insufficient number of factors, future predictions are unreliable because important information is missing from the model. On the other hand, if the model contains too many factors, future predictions are also misleading because random variation particular to the training set has been built into the model. This implies that model size might be inferred

from stopping criteria based on predictive ability. Pirouette makes such an inference when cross-validation is applied during PLS or PCR. We get an idea of how well a regression model performs by using it to make predictions on samples for which we already know “true” values for the dependent variables. Thus, for a validation sample \mathbf{x}_v , we make the following prediction, based on our k factor regression vector β_k :

$$\hat{y}_v = \mathbf{x}_v \beta_k \quad [7.20]$$

and generate the prediction residual:

$$\hat{f}_v = y_v - \hat{y}_v \quad [7.21]$$

where y_v is the “true” value for the dependent variable of the validation sample. To keep the notation simple, hatted symbols (*e.g.*, \hat{y}_v) will indicate a k factor estimate of a quantity.

For a set of n_v validation samples, a Prediction Residual Error Sum of Squares can be calculated *for the y block*:

$$\text{PRESS} = \mathbf{f}^T \mathbf{f} \quad [7.22]$$

Related to the PRESS is the Standard Error of Prediction (SEP, also called the root mean squared error of prediction or RMSEP), which takes into account the number of samples and has the same units as the y variable:

$$\text{SEP} = \left(\frac{\text{PRESS}}{n_v} \right)^{1/2} \quad [7.23]$$

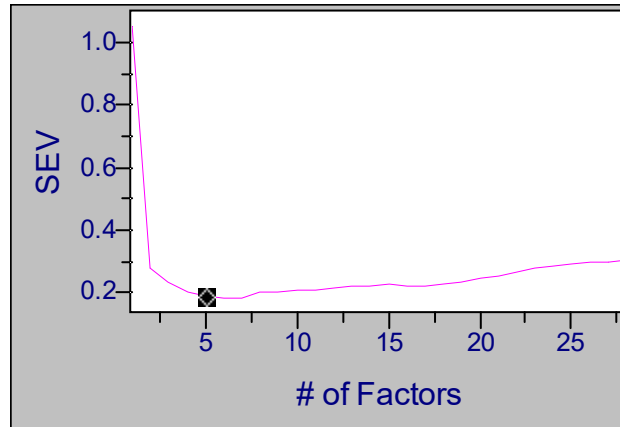
The most naïve version of validation predicts on the training set samples. This type of standard error is termed a Standard Error of Calibration (SEC). The SEC must be corrected for the number of factors k in the model:

$$\text{SEC} = \left(\frac{\text{PRESS}}{n_v - k} \right)^{1/2} \quad [7.24]$$

Of course, the SEC is a tainted measure of future model performance since the regression vector was found by minimizing the PRESS: compare [equation 7.4](#) and [equation 7.22](#). Only if all future samples were exactly like the training set, would the SEC be an useful measure of model reliability. For future samples which are similar, but not identical, to the training set, the SEC is too optimistic, implying smaller prediction errors than will be observed. Moreover, the SEC often decreases steadily with increasing number of factors, which limits its utility as a stopping criterion.

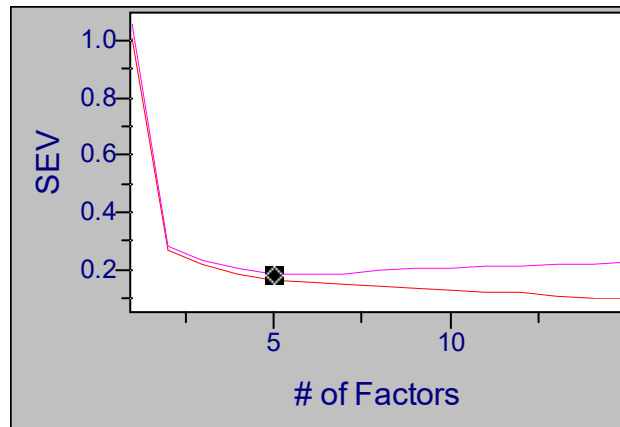
If, however, a model is validated with samples not included in the modeling process, this SEP often shows a different structure from the SEC. A classical SEP is illustrated in [Figure 7.1](#); the optimal number of model factors is indicated by the SEP minimum. Note that when too many or too few factors are retained in the model, the classical SEP increases.

Figure 7.1
A classical SEV plot



A separate data set strictly for validation is not always available. A compromise can be achieved via an internal validation approach, such as Cross Validation. Samples from the training set are temporarily left out and a model created from those remaining. From this model, a prediction of the dependent variable of the left-out samples is made and the y residuals recorded. The left-out samples are then returned to the training set, more are excluded, a new model made, and new predictions and residuals generated. This process is repeated until every sample has been left out once. A PRESS calculated from the accumulated residuals is then converted to a corresponding Standard Error of Cross-Validation (SECV). The SECV, whose denominator is not corrected for the model size, is typically larger than the SEC. It often shows a structure like a classical SEP except that it tends to be somewhat flatter. The next figure compares SEC and SECV.

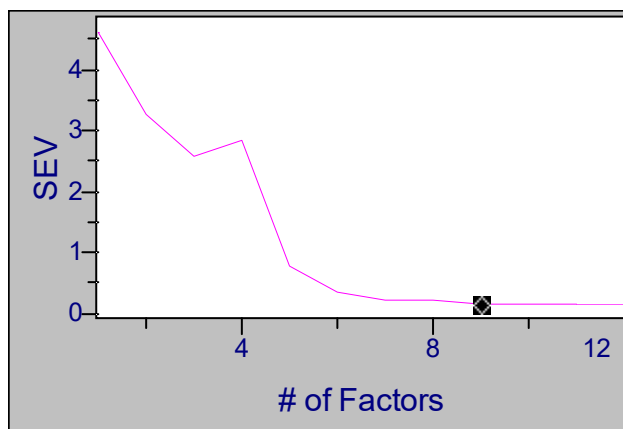
Figure 7.2
SEC (lower trace) and SEV (upper)



Note: Pirouette labels both Standard Error of Cross-Validation plots and Standard Error of Step-Validation plots as SEV. The term SECV is used in these discussions to be more specific and to be consistent with other publications.

The SECV can be used to indicate the optimal number of factors for a regression model so long as the curve exhibits a classical structure. It is a simple matter to find the minimum in a plot of SECV versus number of factors. However, this minimum may not be significantly different from that of a model with 1 fewer component. Figure 7.3 shows a case where 9 factors are extracted but the 8 factor model is not deemed statistically different from one containing 9 factors.

Figure 7.3
SEV plot with a flat
minimum



An F test can determine if two PRESS values are significantly different.⁵⁻⁶ Note that we need only compare those models having fewer factors than the minimum PRESS model. For the typical case of more variables than samples,

$$F_k = \frac{\text{PRESS}_k - \text{PRESS}_{\min}}{\text{PRESS}_{\min}} \frac{n}{n-k} \quad [7.25]$$

with F_k compared against tabulated values using k and $(n-k)$ degrees of freedom and a probability level of 95% (set internally in Pirouette). If there is no significant difference, we choose the more parsimonious model, *i.e.*, the one with fewer factors. When cross-validation is performed in Pirouette, the number of optimal factors is based on such an F test. Like the eigenvalue-based estimate, it is not carved in stone—you may override it.

Note: If X is mean-centered or autoscaled **and** the number of samples is less than the number of independent variables, all occurrences of the term $n-k$ become $n-k-1$.

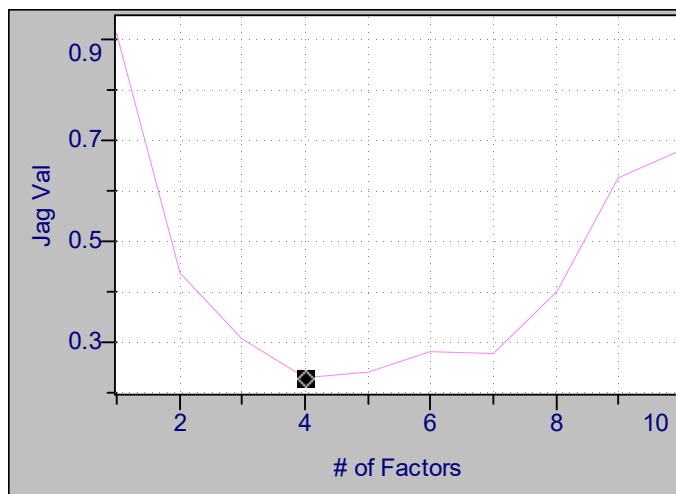
The SEP estimates the precision of future predictions. When you have completed the model building phase (including validation), you must decide if the SEP is small enough to justify using the model. Remember that models are built to replace one set of measurements with another. Suppose the precision of the reference technique (the one you wish to replace) is significantly better than your model's SEP. Can you tolerate this loss of precision? If so, proceed to use the model. If not, put the model aside and consider several possibilities. Perhaps you have not sufficiently optimized the model. Perhaps a change in preprocessing, the application of different transforms, or the exclusion of some variables would decrease the SEP to an acceptable value. On the other hand, the quality of the model data may be insufficient. Perhaps you need greater chromatographic/spectroscopic resolution or more samples. Perhaps the hypothesis that these x measurements can replace these y measurements is just wrong.

Jaggedness

It is wise not to rely on a single diagnostic to determine the optimal number of factors. Many chemometricians recommend looking at the loadings and the regression vector as well. In particular, you can often tell by the characteristics of either of these objects whether there are too many factors in a model: when an individual loading vector or the regression vector starts to appear noisy or jagged is an indication that the model is overfit.

A metric that characterizes jaggedness attempts to quantify the relevance of the noise compared to the overall signal. This measure is developed on the regression vector from either a simple calibration or from a cross-validation⁸. Interpretation of the optimal number of factors based on jaggedness is much like that when using the SEV: find a minimum in the curve. An example jaggedness plot for a cross-validation is shown below.

Figure 7.4
A jaggedness plot



Finding Outliers

Outliers can distort an estimate of the optimal number of factors. If a sample has a unique pattern of variation, an extra factor may be required to fit it! Thus, outliers must be eliminated from the training set before the optimal number of factors can be properly estimated. When no cross-validation is specified, the approach to finding outliers in PCR and PLS is quite similar to that taken in PCA. Below are remarks pertinent to computations made when cross-validation is turned on and descriptions of some outlier diagnostics specific to regression algorithms.

Cross-validated results: Y values, Sample Residuals and Scores

Traditionally, the only result computed for the left-out sample(s) is the predicted y value. In Pirouette, however, the PLS and PCR X residuals and scores are also cross-validated, which improves the reliability of outlier detection during both the model building and prediction phases. This improvement can be understood by recalling that the basic rationale for cross-validation is to calculate certain sample results using a model from which that sample was excluded. Scores of the left-out sample(s) are stored after each pass during cross-validation and figure in the computation of the associated X residuals. Not surprisingly, these (cross-validated) X residuals may be larger than those computed when all samples are included in the model. The resulting (cross-validated) model residual variance bears the same relationship to the unvalidated model residual as the SECV to the SEC, that is, it is the preferred estimate.

Note: *Even though the validated X residuals are computed for each factor during cross-validation, they are not available for display after the algorithm completes due to the storage requirements. Similarly, the validated Scores are not shown. One consequence of this is that a manual calculation of the Mahalanobis distance from the values shown in the Scores object will not produce the values shown in the Outlier Diagnostics object.*

During the modeling phase, the validated Mahalanobis Distance, Studentized Residual and Probability values will differ from the unvalidated versions. In some cases the difference will enhance the detectability of an unusual sample. During prediction a cross-validated regression model will use the cross-validated model residual. Thus, when samples are predicted using the cross-validated model, they may be detected as normal instead of being flagged as unusual when compared to a too small, unvalidated model residual.

Note: *Although the regression vectors for the two PLS flavors (bidiagonalization and NIPALS) will be identical, the X Residuals may differ. The discrepancy arises when the scores and loadings are used to produce an estimate of the original X. The last off-diagonal in the trimmed R matrix (see equation 7.14) is included in the computation in the NIPALS formulation but is not in bidiagonalization⁷. This can influence outlier determinations.*

Leverage

Influential or high-leverage samples are of particular interest when looking for outliers. If a sample's profile differs greatly from the average training set profile, it will have a great influence on the model, drawing the model closer to its location in factor space. A sample's influence is quantified by its leverage, h_i . For the i th sample,

$$h_i = \frac{1}{n} + \mathbf{t}_i^T (\mathbf{T}_k^T \mathbf{T}_k)^{-1} \mathbf{t}_i \quad [7.26]$$

This quantity represents a sample's distance to the centroid of the training set⁹; it is similar to the Mahalanobis distance discussed in "Mahalanobis Distance" on page 5-25. As model size grows, leverage increases, until $h_i = 1$ when $k = g$, *i.e.*, all samples have equal influence over the model. However, when k is less than the maximum, a rule-of-thumb allows us to distinguish unusually influential samples:

$$h_{\text{crit}} = \frac{2k}{n} \quad [7.27]$$

Samples with leverages greatly exceeding h_{crit} should be examined closely. Keep in mind, however, that influential samples are not necessarily outliers. If a valid sample lies a large distance from the center of the training set because it has an extreme value of the dependent variable, it contributes important information to the model.

Studentized Residuals

It is natural to examine Y residuals when looking for outliers. However, a sample's raw y residual is misleading due to the effect of leverage. If a sample's y value is extreme, it has a greater influence on the model than a sample close to \bar{y} . The extreme sample "pulls" the model toward it, decreasing the difference between the observed value of y and its fitted value. In contrast, a sample lying close to \bar{y} , having little or no leverage, cannot influence the model so its residual tends to be larger. The Studentized residual takes leverage into account, thus giving a fairer picture of differences among residuals. It is derived from the root mean squared residual for the training set¹⁰:

$$\text{RMSE} = \left(\frac{1}{n-k} \sum f_i^2 \right)^{1/2} \quad [7.28]$$

The Studentized residual, r_i is then:

$$r_i = \frac{f_i}{\text{RMSE}(1 - h_i)^{1/2}} \quad [7.29]$$

Because it is assumed that r_i is normally distributed, a t test can determine whether a sample's Studentized residual is "too large". In Pirouette, we compute a value for r_{crit} at a 95% probability level (set internally), based on the n training set samples.

Predicting in PCR/PLS

Predictions are made in PCR/PLS by multiplying the x block of a new sample as shown in [equation 7.20](#) using the k factor regression vector found during model optimization to produce an estimate of y . It is also helpful to know if the new sample differs significantly from the training set. This decision is based mainly on the magnitude of the x residuals and scores when the new sample is projected into the model factor space: samples significantly different will have large residuals and scores. Two other previously mentioned quantities, Mahalanobis distance and Leverage, have prediction analogs. The former is described in "[Mahalanobis Distance in Prediction](#)" on [page 5-31](#). The latter is defined in [equation 7.26](#).

Prediction Confidence Limits

After prediction we might also like to know the level of uncertainty in the predicted property or concentration. If we include cross validation during calibration or if we run a prediction on a data set containing known Y values, the resulting SECV or SEP gives an idea of the **overall** expected error level. However, this value is a constant and will not give a meaningful measure of the uncertainty for individual samples.

Instead, we can compute a sample-specific confidence limit¹¹ based on the model parameters and on h , the sample's leverage (see "[Leverage](#)" on [page 7-11](#)).

$$\text{CL} = t \cdot \text{SEC} \cdot \sqrt{(1 + h)} \quad [7.30]$$

where the t is from Student's distribution based on the prediction probability setting (see "[Prediction](#)" on [page 10-19](#)) and the degrees of freedom in the model—the number of samples reduced by the number of model factors, and the SEC is the standard error calibration (see [page 7-7](#)).

ORTHOGONAL SIGNAL CORRECTION

Although factor based regression methods are very good at isolating relevant information in a data matrix within just a few factors, the irrelevant information—that which does not contribute to the correlation with the dependent variable—may still be confounding. Subsequent regression models may not be as robust and their interpretation may be confusing.

The goal of the calibration is to segregate that portion of the X block that is correlated to the Y variable from the portion that is not correlated. Thus, orthogonal signal correction (OSC) was developed to remove components in the X block that are orthogonal to the Y block before performing the regression. The OSC-corrected X block should then produce models which are more parsimonious and/or result in lower prediction errors.

In Pirouette, the method of direct OSC¹² is offered as an enhancement to PLS and PCR.

Mathematical Background of DOSC

The OSC method uses several projections to segregate correlated and uncorrelated information. First is a projection of y onto X :

$$\hat{y} = \mathbf{X}\mathbf{X}^\dagger y \quad [7.31]$$

where the dagger superscript indicates the pseudoinverse. The X matrix is then projected onto this \hat{y} in a similar way:

$$\hat{X} = \hat{y}\hat{y}^\dagger X \quad [7.32]$$

After deflation, that portion of X which is orthogonal to y remains:

$$\mathbf{X}_{oy} = \mathbf{X} - \hat{X} \quad [7.33]$$

This matrix can be decomposed into scores T and loadings P :

$$\mathbf{X}_{oy} = \mathbf{T}\mathbf{P}^T \quad [7.34]$$

The scores are then trimmed to the number of orthogonal components to be removed from X . Orthogonal weights can be obtained from a direct multiplication of the inverse of X_{oy} and the scores. However, it has been found¹² that the slightly inexact generalized inverse works better than the pseudoinverse, and this is designated by X^{-1} .

$$\mathbf{W}_{osc} = \mathbf{X}_{oy}^{-1} \mathbf{T} \quad [7.35]$$

From W , the orthogonal scores can be computed,

$$\mathbf{T}_{osc} = \mathbf{X}\mathbf{W}_{osc} \quad [7.36]$$

and the corresponding orthogonal loadings.

$$\mathbf{P}_{osc} = \mathbf{X}^T \mathbf{T}_{osc} (\mathbf{T}_{osc}^T \mathbf{T}_{osc})^{-1} \quad [7.37]$$

Finally, the orthogonalized X matrix results from removing the orthogonal signal from X .

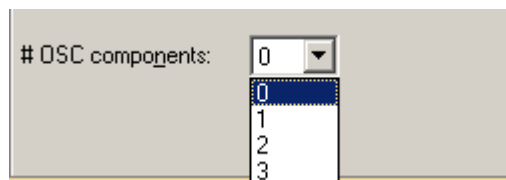
$$\mathbf{X}_{osc} = \mathbf{X} - \mathbf{T}_{osc} \mathbf{P}_{osc}^T \quad [7.38]$$

It is this matrix X_{osc} that is substituted for the normal X matrix in the subsequent PLS or PCR analysis.

Using OSC with PLS/PCR

As many as three OSC components can be specified in the Run Configure dialog.

Figure 7.5
Setting the number
of OSC components



Sufficient information is stored in a Pirouette model to allow the removal of the same OSC components during prediction.

RUNNING PCR/PLS

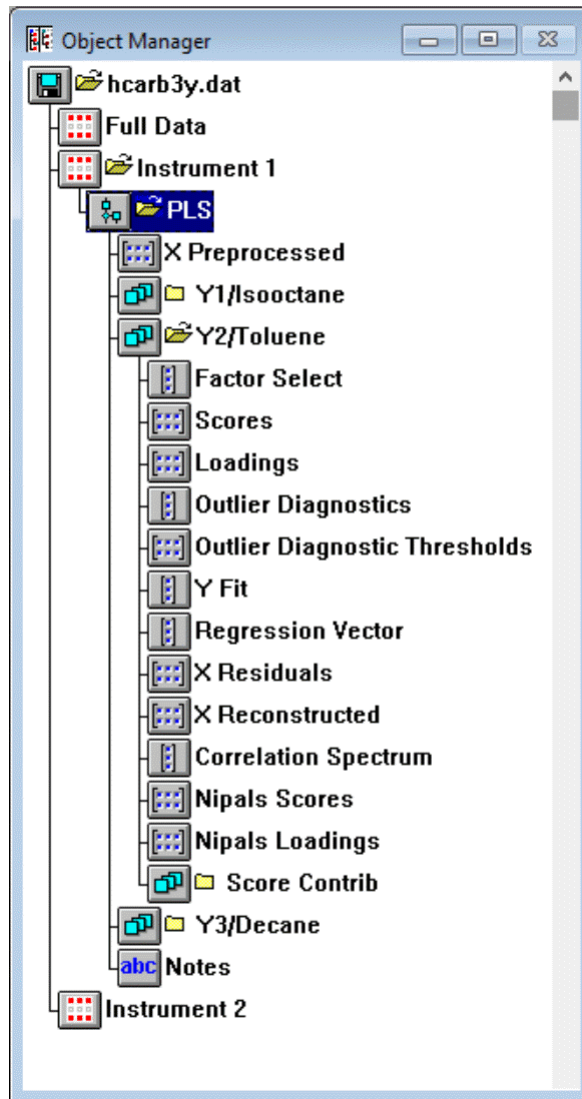
The options associated with these two factor based techniques are described in [“PCR and PLS Options” on page 16-24](#). When the PCR or PLS algorithm executes, many computed objects are created and displayed. The objects computed during factor-based regression can help you find sample outliers, choose the optimal number of factors, and make decisions about excluding variables. Each is described below along with ideas about how to examine them.

In addition to the computed objects, information necessary to make predictions for each dependent variable included in the training set is stored in memory as pieces of a regression model. A model can be used as soon as it has been created, or it can be stored separately from the training set data and reloaded later to make predictions on future samples. A Pirouette regression model is more than just a regression vector based on k factors. It also contains information about which variables were excluded and what transforms/pre-processing options were chosen so that future samples are treated in the same way as the training samples. Model building is an iterative process. You will seldom run a regression algorithm just once and immediately start making predictions. Instead you will spend much of your time optimizing your model, that is, finding the “best” set of samples, variables and configuration parameters.

As mentioned previously, PCR and PLS produce analogous results. Two objects are associated with PLS only, the NIPALS scores and loadings discussed in [“NIPALS Scores and Loadings” on page 7-6](#). As shown in [Figure 7.6](#), each dependent variable has its own set of objects.

Different objects default to different views. You can change an object view with the various View buttons. Many objects are matrices whose component vectors can be accessed via the Axes Selector button. You should experiment freely with different object views and arrangements. Particularly informative views and arrangements are captured and described in the figures which accompany the following discussion.

Figure 7.6
Object Manager
listing of PLS results

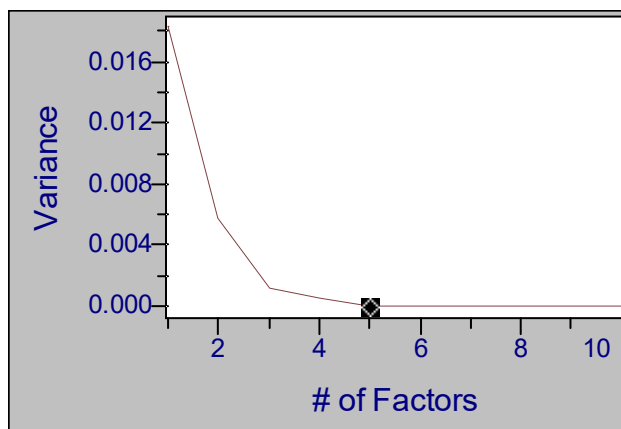


Factor Select

The Factor Select object holds measures of the amount of variation captured by each factor extracted during PCR/PLS regression and standard error calculations.

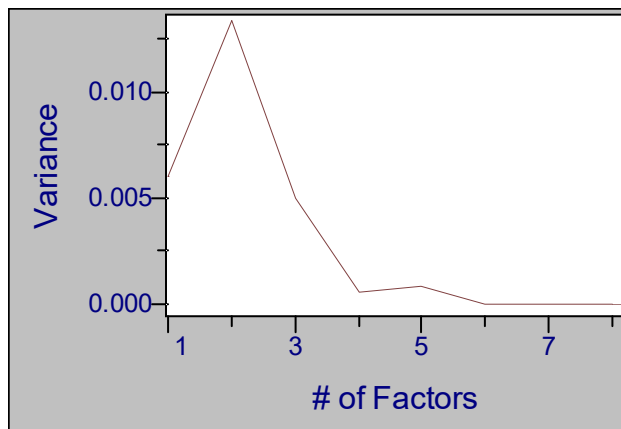
The first three columns are derived from the eigenvalues. The first column, labeled variance, holds the eigenvalues. Percent variance and cumulative percent variance are supplied also. PCR eigenvalues decrease monotonically. This is a consequence of the PCA decomposition: each successive factor accounts for a smaller amount of variation than the previous one. In fortunate cases, the decrease in eigenvalues is precipitous and the boundary between relevant and irrelevant factors is obvious. A gradual dropoff with no clear break, as seen in [Figure 7.7](#), makes it difficult to perceive any boundary.

Figure 7.7
Eigenvalues plot
with gradual
decrease



As the first few PLS factors are extracted, the eigenvalues may decrease. However, later PLS eigenvalues often increase as shown in Figure 7.8. This behavior occurs because the PLS algorithm extracts factors in order of decreasing correlation between the X block / Y variable correlation. Thus, PLS eigenvalues are **not** helpful in determining the optimal number of factors.

Figure 7.8
PLS eigenvalues plot



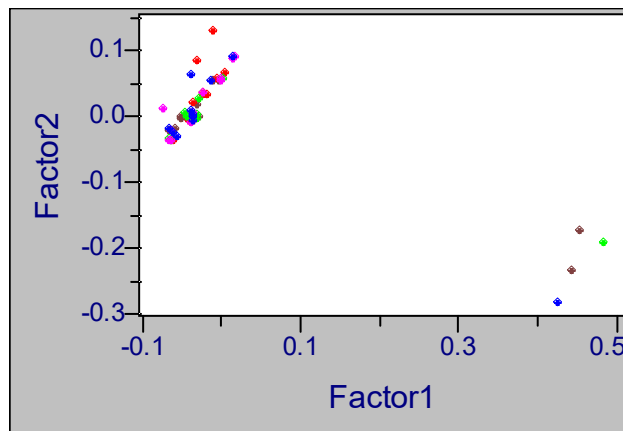
Factor Select also contains: (1) RMSEC which is in the units of the measurement, (2) Press Cal, the PRESS based on the training set, (3) r Cal, the linear correlation coefficient relating the predicted and measured values of the dependent variable, and (4) Jag Cal, the Jaggedness calculation for the calibration. If validation is specified, the object also contains the analogous validation quantities: r Val, Press Val, and SEV. Without validation, the Variance (*i.e.*, the eigenvalues) trace is shown initially; with validation, SEV is shown initially. The SEC, SEV and PRESS quantities are defined in [“Validation-Based Criteria”](#) on page 7-6.

Regardless of which trace or combination of traces is plotted, a diamond is always available on one curve, indicating Pirouette’s choice of the “optimal” number of model factors. For an explanation of how Pirouette arrives at this number for validated models, see [“Validation-Based Criteria”](#) on page 7-6. For unvalidated models, the approach is outlined in [“Estimating the Number of Factors in Unvalidated Models”](#) on page 5-21. Change the number of factors by clicking above the desired x axis value. Many regression objects are a function of this setting; changing the position of the diamond marker triggers a recalculation of these so-called linked objects. Part of model optimization consists of examining factor dependent objects while changing the diamond position.

Scores

PCR/PLS score plots show data set homogeneity and draw attention to unusual samples or groupings in the data; see “Scores” on page 5-35. In an ideal training set used for regression modeling, the scores plot contains a cloud of points with no clustering and no sparse regions. Clustering in the scores suggests either inhomogeneous sampling or a need for more than one regression model. In Figure 7.9 an inhomogeneous data set is shown; two groups can be identified.

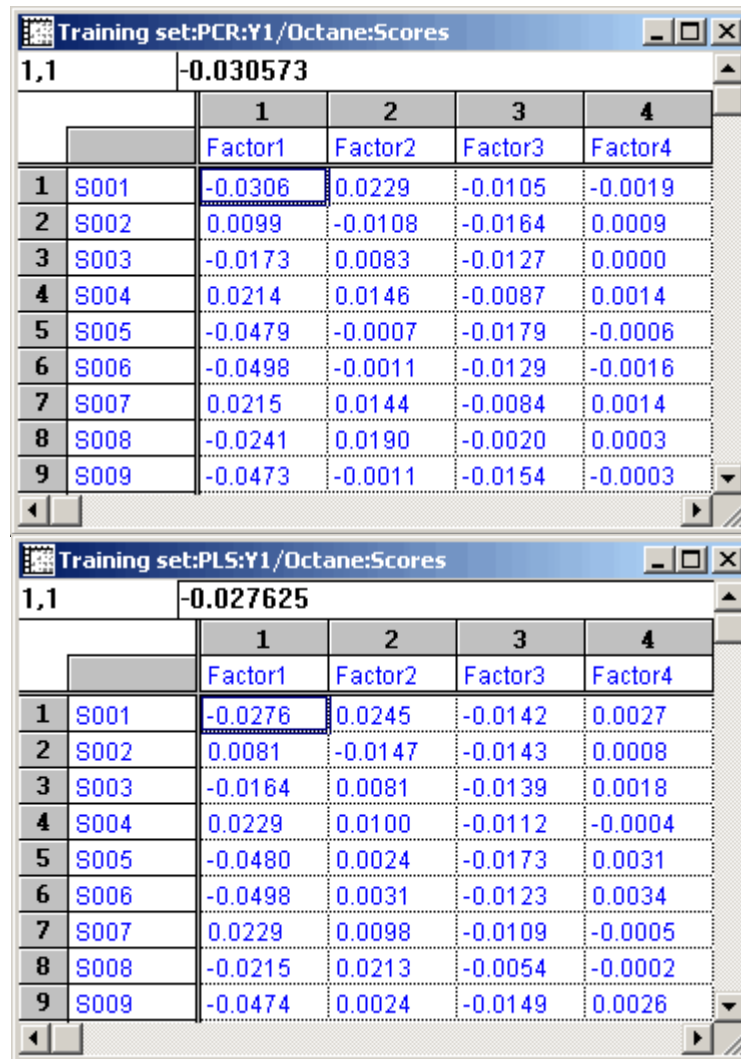
Figure 7.9
Score
inhomogeneity



As mentioned previously, scores are computed differently for PLS and PCR. Because PCR starts with a PCA decomposition on the X block, PCR and PCA scores are identical although some small discrepancies may arise from algorithm differences (SVD for PCR versus NIPALS for PCA). Because PCR scores are derived strictly from the X block, they do not vary with the y variable. PLS scores, however, are different for each y since Y block variation plays a part in the PLS decomposition. See Figure 7.10 for a comparison of PCR and PLS scores.

Note: *Because PLS scores vary with dependent variable, you must examine the PLS scores object associated with each dependent variable. The scores for one dependent variable may look homogeneous while another may have obvious sample groupings.*

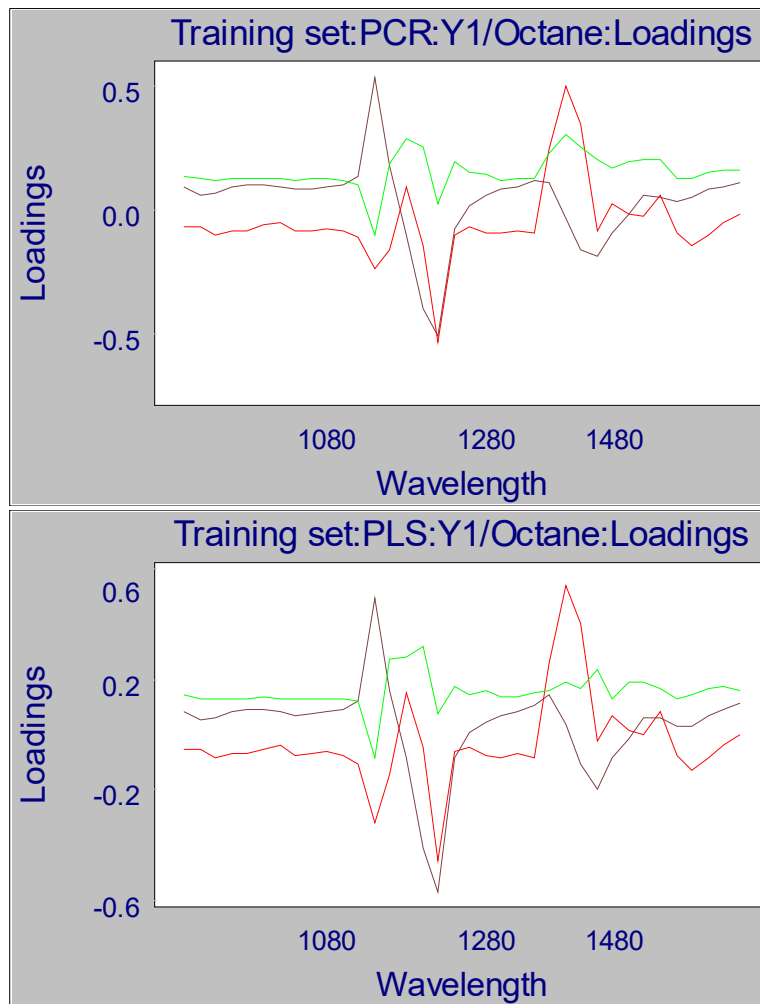
Figure 7.10
Comparing PCR and
PLS scores



Loadings

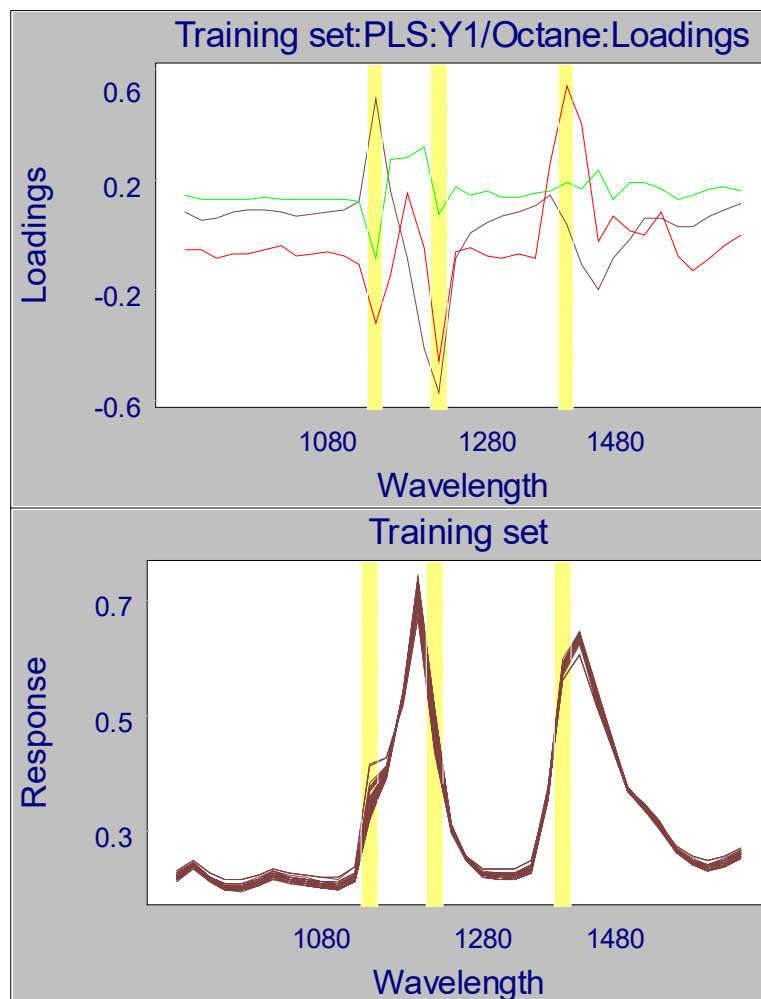
Loadings reveal how the measured variables combine to form the factor axes; see “Loadings” on page 5-36 and in Figure 5.21 for more details. A loadings plot for spectroscopy data is shown in Figure 7.11.

Figure 7.11
Loadings plots:
(a) for PCR;
(b) for PLS



Comparing a line plot of the loadings and the original data helps us see what data features are being modeled by each factor. This may tell us something about what phenomena are driving the model. Such a comparison is shown below where features apparent in the loadings plot have been highlighted. Note that features in the loadings may not necessarily correlate to peaks in the spectra.

Figure 7.12
Comparing loadings
to original data
showing important
features

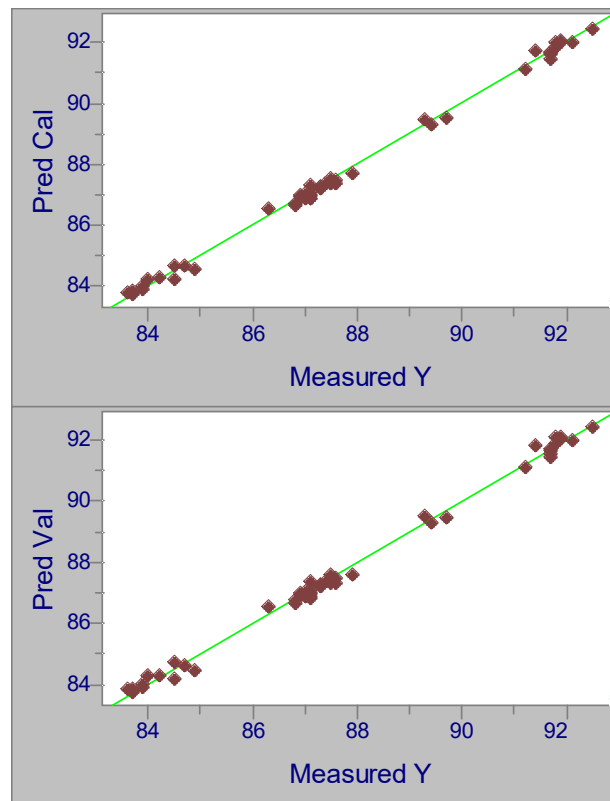


When deciding on the optimal number of factors to retain in a model, consider excluding factors which have noisy loadings. For spectroscopic data, smooth spectra should give rise to smooth loadings as long as non-random variation is being modeled.

Y Fit

The Y Fit object holds the results of predicting on the training set samples. With no validation, Y Fit consists of three vectors: (1) the measured y value, Measured Y, which is merely a copy of the dependent variable; (2) the predicted y value, Pred Cal; and (3) the y residual, Res Cal. With validation, Pred Val and Res Val, the validation analogs of Pred Cal and Res Cal, are included. The default view of Y Fit is a 2D plot of Pred Cal (if no validation) or Pred Val (if cross or step validation) vs. Measured Y. The next set of figures shows both cases. The number of model factors is displayed in the lower right corner of each plot.

Figure 7.13
 Y Fit default view (a)
 without validation (b)
 with validation

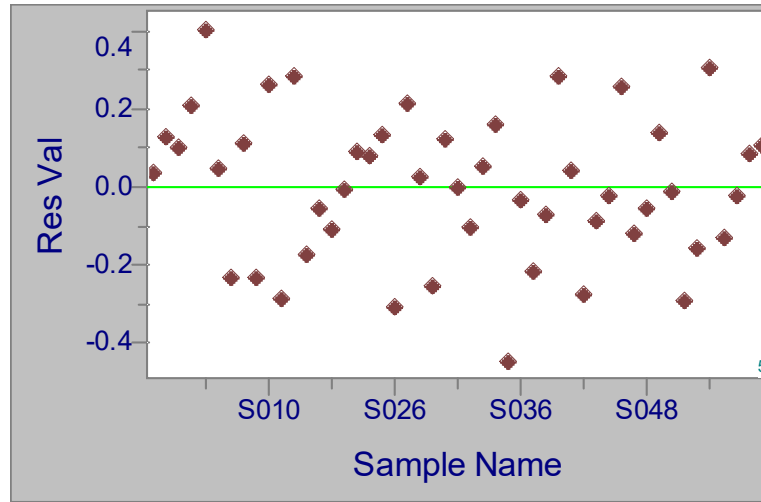


When two Y Fit vectors are plotted against each other in a 2D view, appropriate reference lines are also displayed. Plots of a residual vector contain horizontal or vertical zero reference lines; plots with only predicted or measured Y values have a diagonal reference line of slope one, as in the previous figure.

Plotting Pred Cal or Pred Val against Measured Y gives a sense of the respective descriptive and predictive model quality. If the model perfectly describes the relationship between the X block and Y variable, all sample points fall on the diagonal. However, when the optimal number of factors is less than the maximum allowed, sample points scatter around the reference line.

A plot of either Res Cal or Res Val against Measured Y (or sample #) can be used to detect trends in the residuals. When inspecting these plots, keep in mind the usual assumptions about residuals: they should be randomly distributed, have constant variance and should not correlate with Measured Y or sample #. If there is evidence of heteroscedasticity (non-constant variance), the assumption that the dependent variable is a linear function of the independent variables may not be warranted. Figure 7.14 shows a case in which the errors do not appear to be a function of sample number.

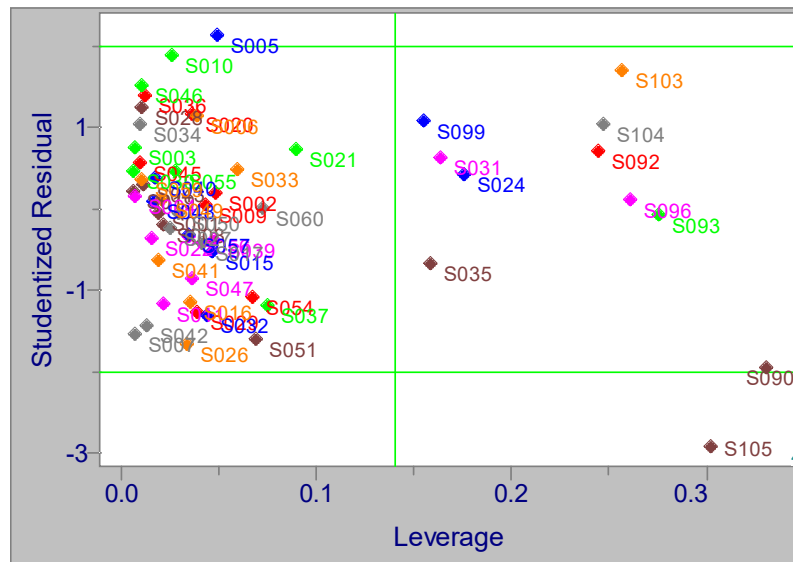
Figure 7.14
Validation residuals
as a function of
Sample



Outlier Diagnostics

The Outlier Diagnostics object consists of Leverage (see [page 7-11](#)), Studentized Residuals (see [page 7-11](#)), Mahalanobis distance, F-Ratio and Probabilities. These last two quantities are discussed in “Mahalanobis Distance” on [page 5-25](#) and “Probability” on [page 5-25](#), respectively. The default view of this object, Leverage vs. Studentized Residual, is shown in [Figure 7.15](#).

Figure 7.15
Studentized
Residual vs.
Leverage



Note: *If cross-validation was performed, this object contains validated quantities. Thus, the Mahalanobis distance is based on left-out scores and the residuals derived computations are based on left-out X residuals.*

As its name implies, the object attempts to identify outliers. The number of factors in the model appears in the lower right corner and approximate threshold lines are supplied. Remember that the thresholds are based on a 95% confidence level so 5% of the population is expected to exceed the cutoff value. Not surprisingly, then, in large data sets, many

samples lie beyond the threshold. Consider eliminating samples which exceed both thresholds or samples which greatly exceed one threshold, then re-run the regression algorithm. If high leverage samples are associated with extreme y values, they may not be problematic, but are merely located at the edge of the calibration range. When you have an approximate idea of model size, check the Outlier Diagnostics for samples which greatly exceed the residual threshold for one size but are within it when an additional factor is extracted. This could indicate that the additional factor is modeling those samples only. Finally, examine the X Residuals of suspect samples, looking for regions of bad fit. This may help you understand in what way they are unlike other training set samples.

An example of a very high leverage sample is shown below; the spectrum is unusual but its associated Y value is not extreme. Such an unusual data point will have a large influence on the regression vector.

Figure 7.16
Sample with high leverage, two views

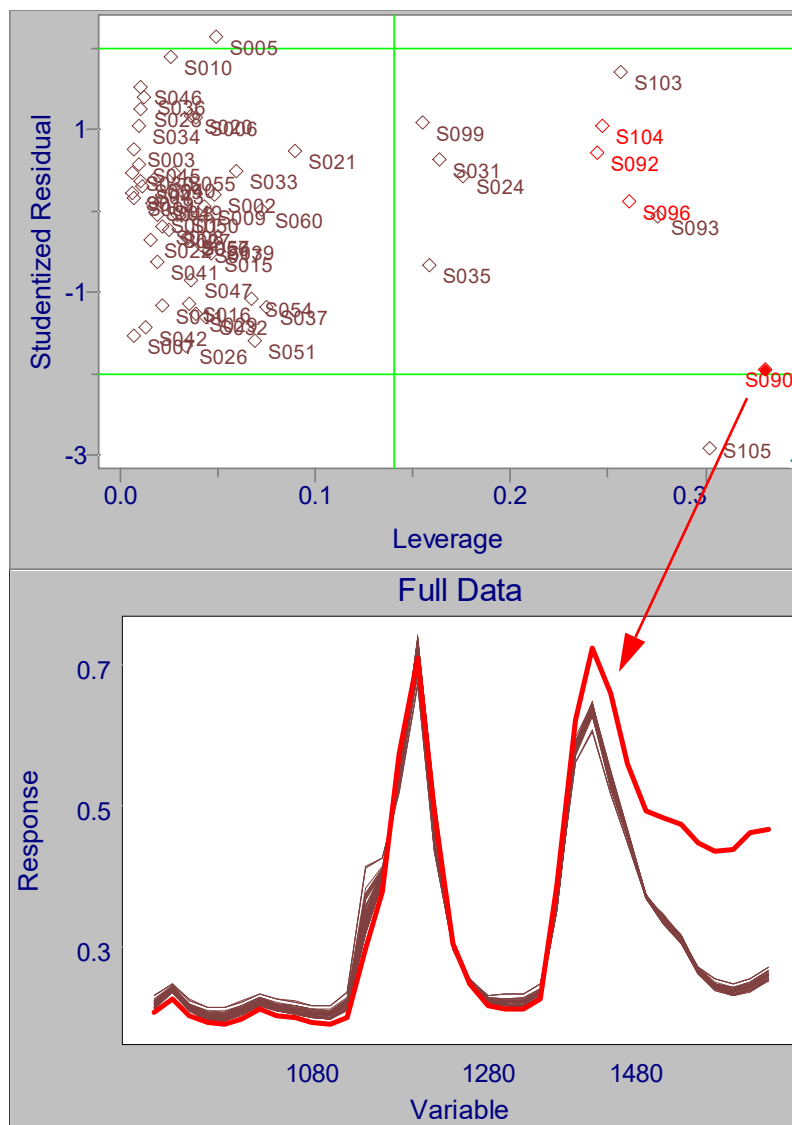
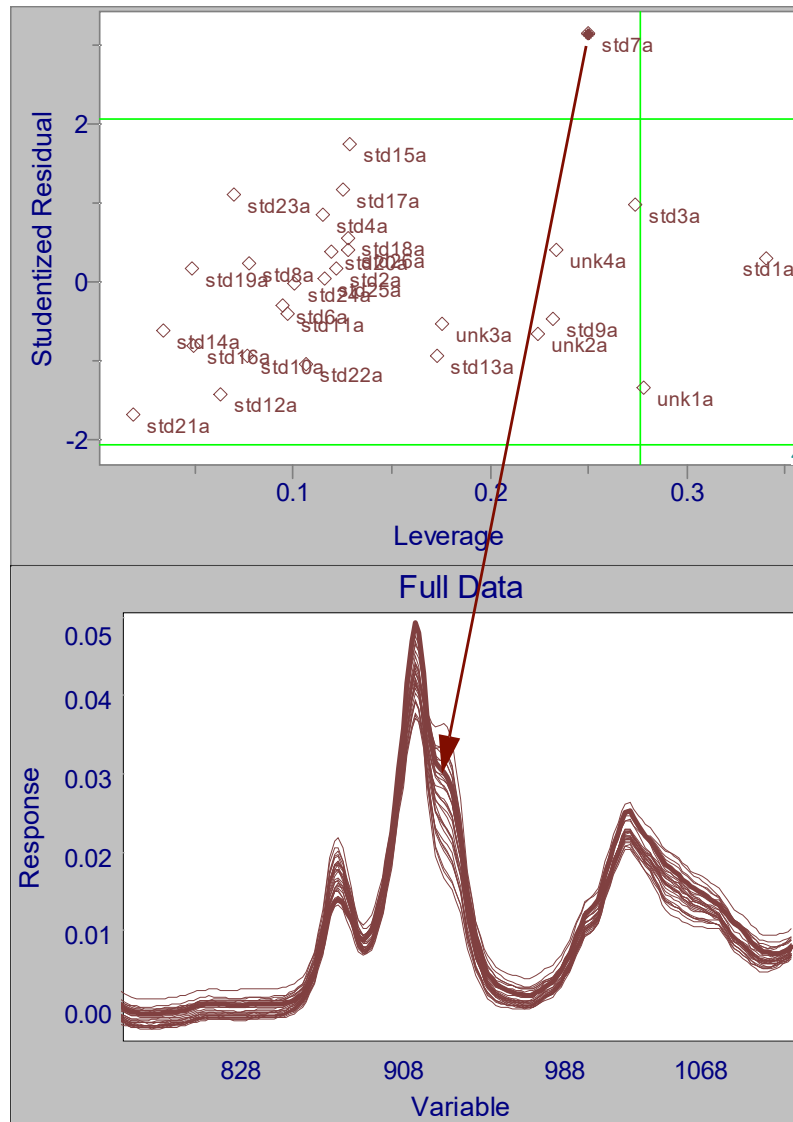


Figure 7.17 shows an example of a sample (std7a) with a high Studentized residual, but normal leverage. The predicted Y value is quite different from the Measured Y although the spectrum looks reasonable.

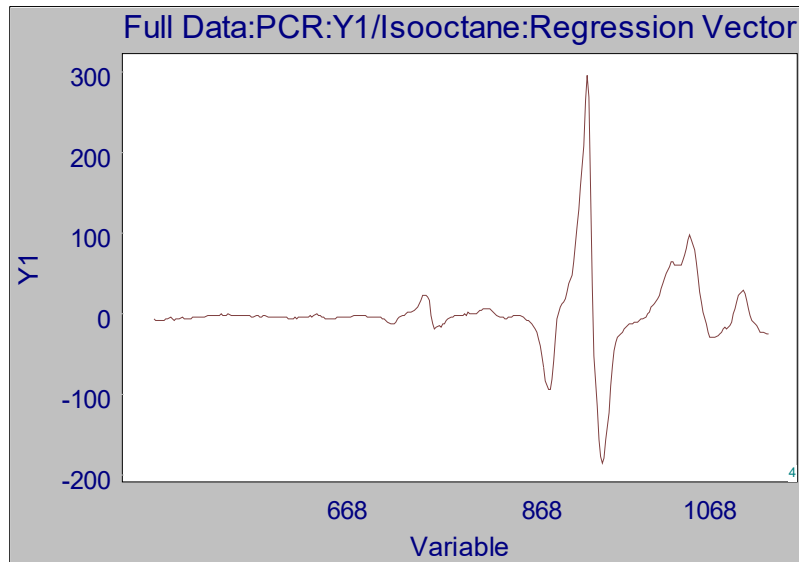
Figure 7.17
Sample with High Studentized Residual; two views



Regression Vector

The regression vector can be thought of as a weighted sum of the loadings included in the model. An example is shown in [Figure 7.18](#). A line plot of this object reveals which independent variables are important in modeling the dependent variable. Variables with very small coefficients do not contribute significantly to a prediction. This information might guide you in eliminating variables.

Figure 7.18
Regression Vector



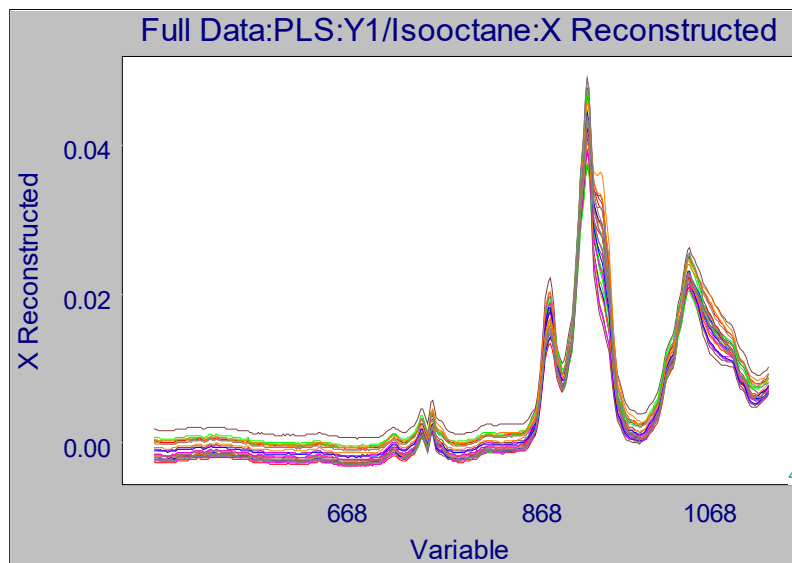
Viewing a line plot of the regression vector as the number of factors changes can be quite instructive. When the number of factors is small and each additional factor accounts for significant variation, the vector's shape changes dramatically with number of factors. Often a point is reached where the changes are much less striking and appear random, signaling that noise is being modeled.

A 2D plot of the regression vector provides a graphical means of excluding variables. See "Correlation Spectrum" on page 7-26 for a description of how to make exclusions using this method.

X Reconstructed

This object is described in "X Reconstructed" on page 5-37.

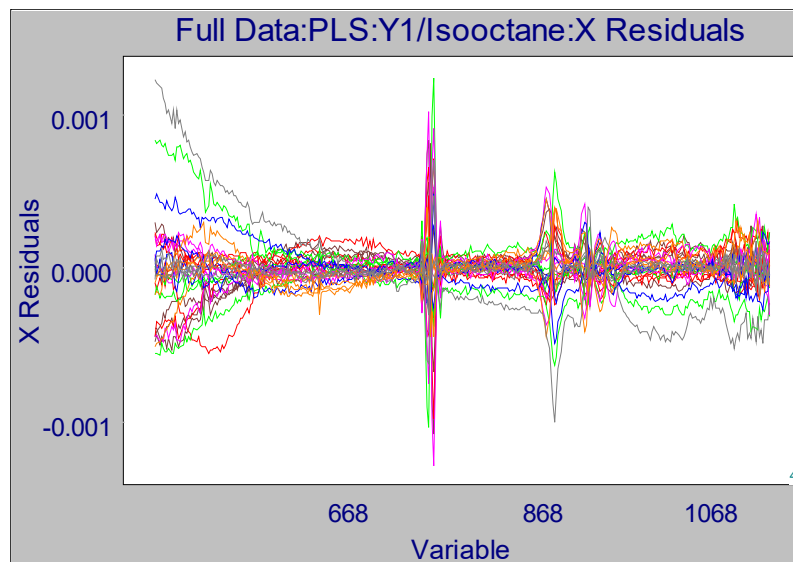
Figure 7.19
X Reconstructed
example



X Residuals

The X Residuals object is described in more detail in “X Residuals” on page 5-38. By default, a line plot of all sample residual vectors is displayed. Viewing this plot as the number of factors is changed may reveal poorly fitting samples/variables. The following figure shows a case where several samples have large residuals around 870 nm for a four factor model.

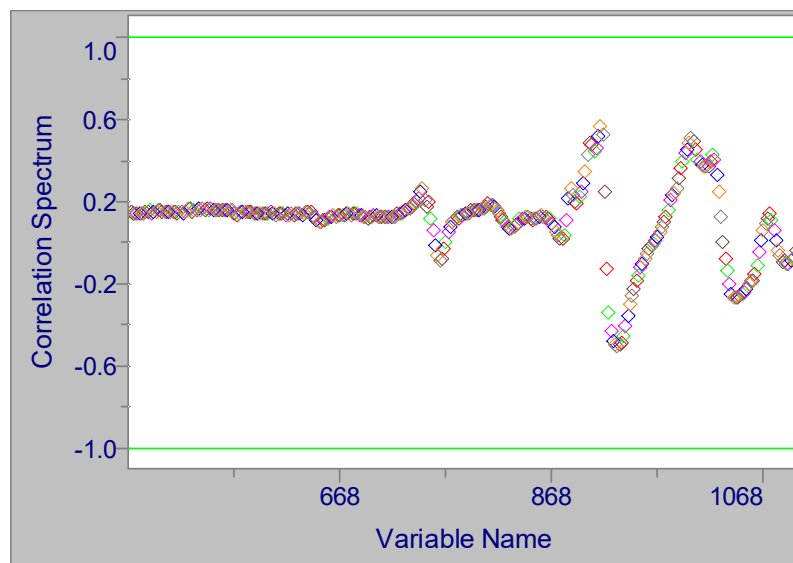
Figure 7.20
X Residuals example



Correlation Spectrum

When a y variable is available, as is always the case in regression problems, it is possible to compute its correlation to each variable in the x block. The Correlation Spectrum is the resulting vector of r values. This object is computed for each y in the training set whenever PLS or PCR is run. A plot of the Correlation Spectrum is shown below.

Figure 7.21
Correlation Spectrum



An interesting function of this plot is to facilitate graphical exclusions of poorly correlating x variables. To accomplish this, click on the Pointer tool and click-drag a rubber box around points with small r values. They become selected (i.e., highlighted) when the mouse button is released. Then choose the Create Exclude entry from the Edit menu to generate a subset which does not contain these variables.

NIPALS Scores

This object is computed only for PLS. The NIPALS algorithm produces PLS scores which differ slightly from those computed by the bidiagonalization algorithm as discussed in “NIPALS Scores and Loadings” on page 7-6. The following graphic shows the scores for the first three factors for both algorithms.

Figure 7.22
Comparison of scores:
(a) from bidiagonalization procedure;
(b) from NIPALS procedure

1,1		0.001740		
		1	2	3
		Factor1	Factor2	Factor3
1	smp1	0.0017	-0.0232	-0.0049
2	smp2	-0.0016	-0.0090	0.0266
3	smp3	0.0171	-0.0136	0.0057
4	smp4	0.0176	-0.0020	-0.0196
5	smp5	0.0207	-0.0625	0.0501
6	smp6	0.0142	-0.0075	-0.0127
7	smp7	0.0033	-0.0037	0.0265
8	smp8	0.0053	-0.0270	0.0412
9	smp9	-0.0011	0.0037	-0.0171
10	smp10	0.0184	-0.0133	0.0145

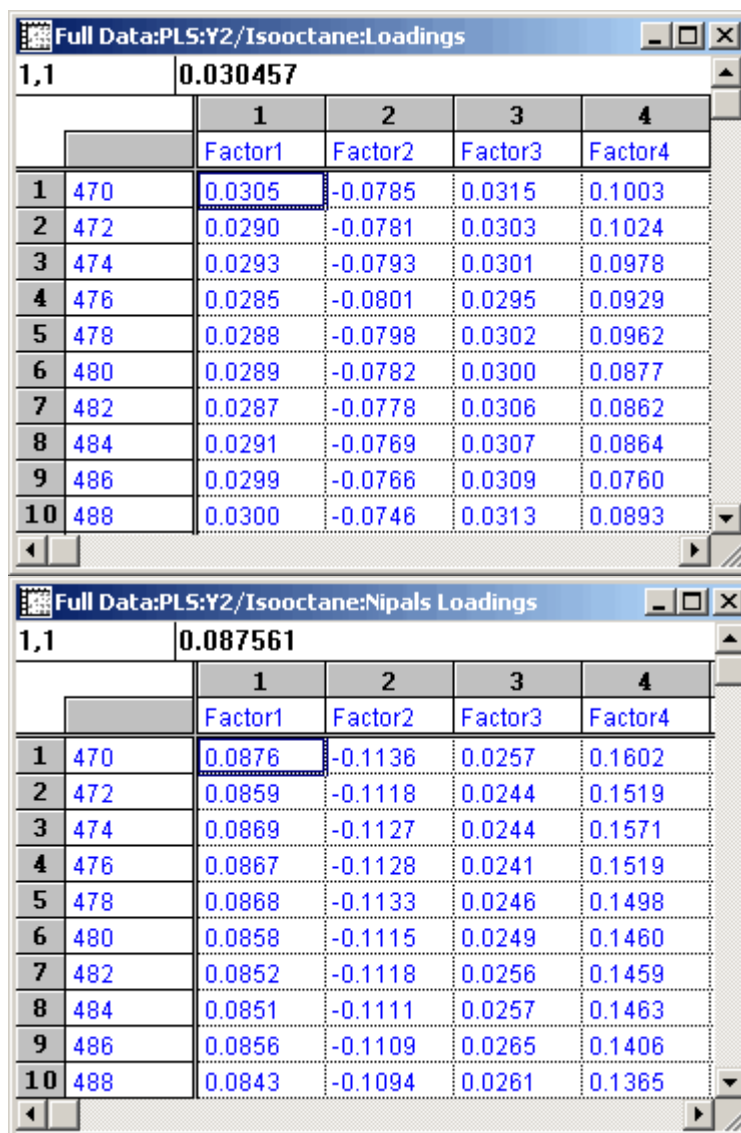
1,1		0.001740		
		1	2	3
		Factor1	Factor2	Factor3
1	smp1	0.0017	-0.0219	-0.0293
2	smp2	-0.0016	-0.0102	0.0152
3	smp3	0.0171	-0.0012	0.0044
4	smp4	0.0176	0.0109	-0.0075
5	smp5	0.0207	-0.0474	-0.0026
6	smp6	0.0142	0.0028	-0.0095
7	smp7	0.0033	-0.0013	0.0250
8	smp8	0.0053	-0.0232	0.0155
9	smp9	-0.0011	0.0029	-0.0138
10	smp10	0.0184	0.0001	0.0146

NIPALS Loadings

The object is computed only for PLS. The NIPALS algorithm produces PLS loadings which differ slightly from those computed by the bidiagonalization algorithm as dis-

cussed in “NIPALS Scores and Loadings” on page 7-6. The graphic below shows a comparison of the first four loadings which is analogous to Figure 7.22.

Figure 7.23
Comparison of loadings:
(a) from bidiagonalization procedure;
(b) from NIPALS procedure



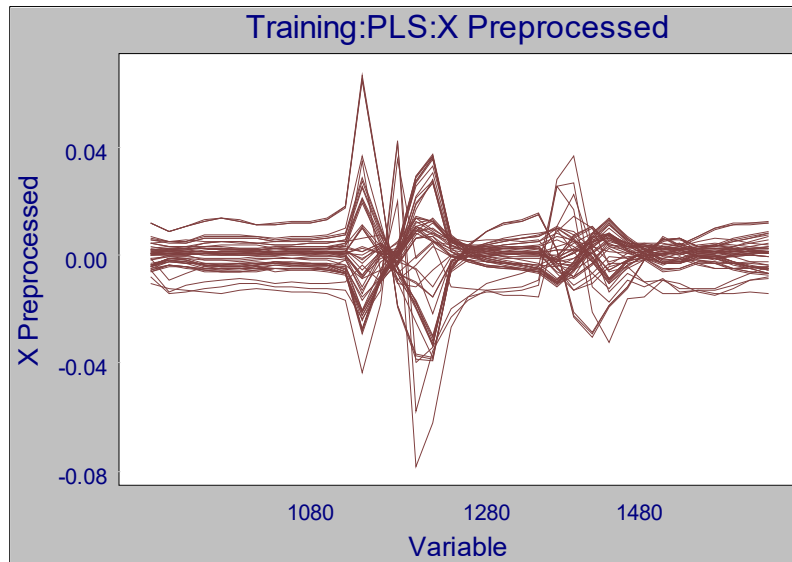
X Preprocessed

As mentioned in “X Preprocessed” on page 5-32, it is often useful to view the X block data after preprocessing. This object is available in PLS, PCR, and PLS-DA.

OSC Results

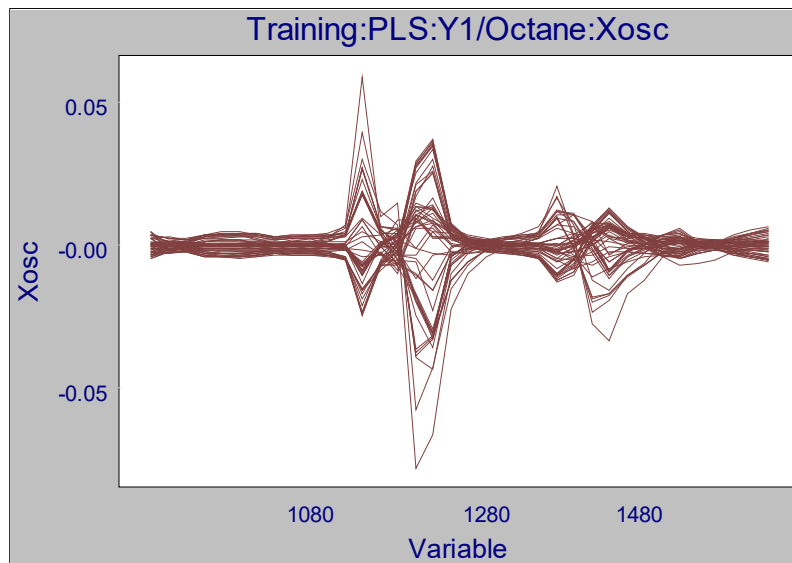
Because orthogonal signal correction is performed on the pre-treated preprocessed X block, it is useful to compare before and after correction. The following figure is from the data in Figure 7.17 and show the profiles after mean-centering.

Figure 7.24
Preprocessed data



These data retain information in the preprocessed X block which are not correlated to the Y block. After removing 1 OSC component, the majority of the uncorrelated information is removed. Figure 7.25 shows the effects of the orthogonalization. The remaining information should produce a more reliable calibration.

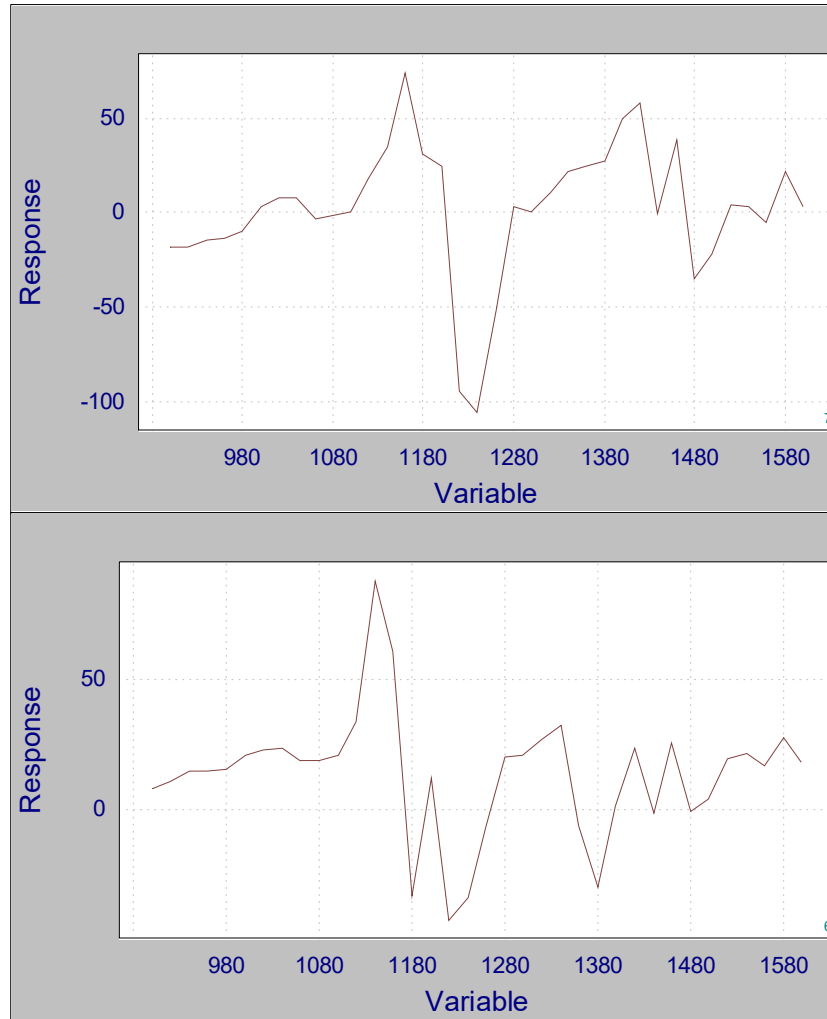
Figure 7.25
Orthogonalized data



After running PLS or PCR with OSC enabled, you should look at the regression vector to see how this change effects interpretation of the model. For example, the following figure contrasts the regression vectors for the data in Figure 7.24.

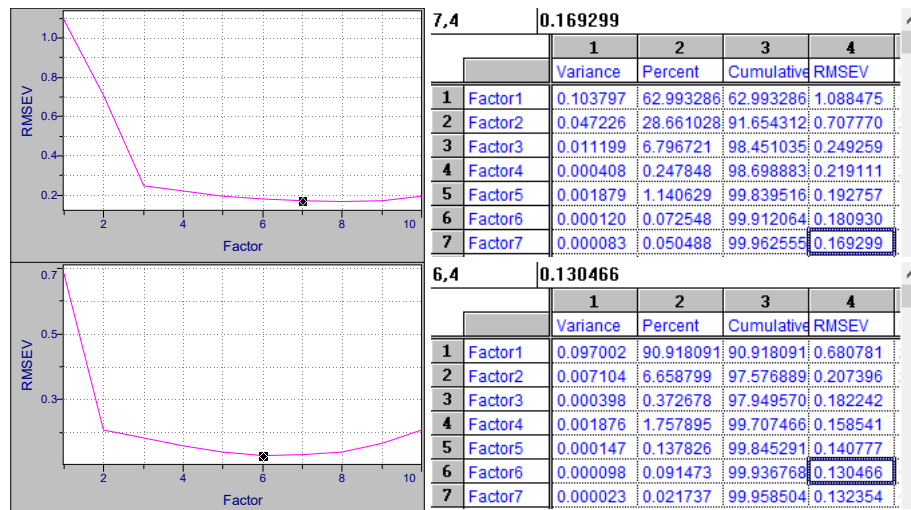
7 Regression Methods: Factor Based Regression

Figure 7.26
Regression vectors
without and with
OSC



In this example, the optimal number of factors was reduced by only 1, but the error (based on the RSECV) was improved by about 20% (see next figure).

Figure 7.27
Standard errors
without OSC (top)
and with OSC
(bottom)



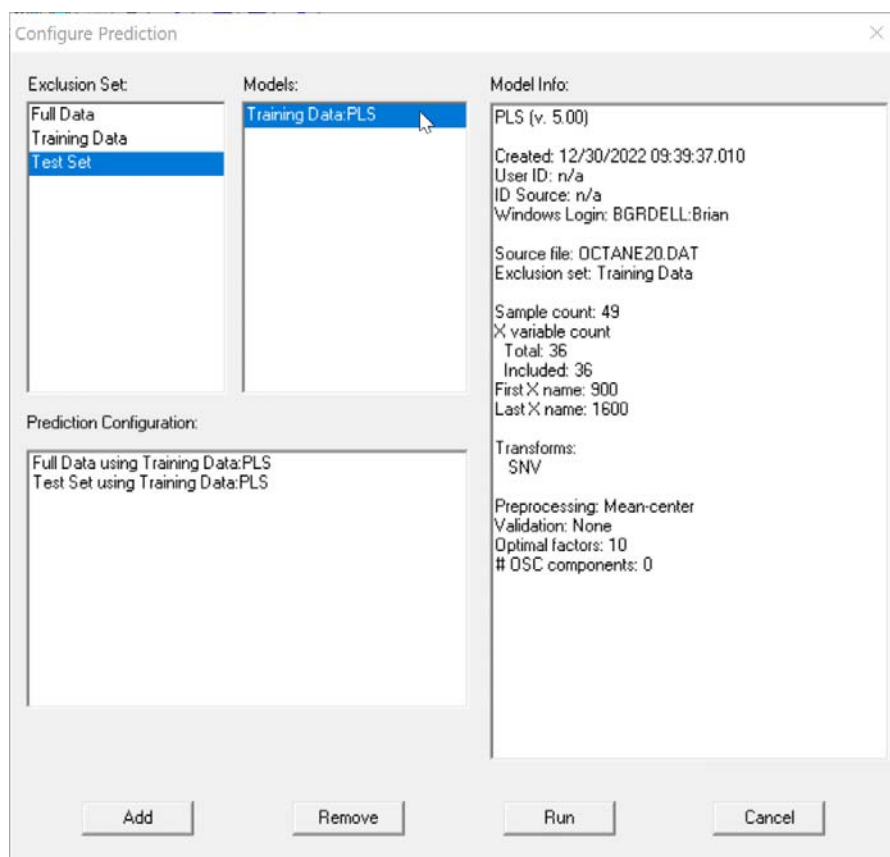
Depending on the data set, removal of more than 1 OSC component may be worthwhile. Thus, varying this parameter becomes **another** set of investigations for model optimization.

MAKING A PCR/PLS PREDICTION

Running PCR/PLS triggers the creation of a model. You can confirm this by going to Process/Predict after running either algorithm and noting the entry under Model. Making predictions requires a model and a target, that is, a data set with an X block containing one or more samples whose y values will be predicted. This data set's X block must have the same number of independent variables as the data set from which the model was created and cannot contain any excluded independent variables. The prediction target may or may not contain dependent and class variables.

Figure 7.28 shows the Configure Prediction dialog box. To get model information, highlight its entry as illustrated in the figure. To configure a prediction, highlight a model and an exclusion set and click on Add. You can configure more than one prediction at a time by highlighting a different model name or exclusion set then clicking Add. Predictions are made when you click Run. Some prediction objects summarize results across dependent variables; others are stored in a folder with the dependent variable name.

Figure 7.28
Configure Prediction
dialog box



Error Analysis

If the prediction target contains data for any dependent variables having the same name as the dependent variables in the training set, the Error Analysis object is created. For each matching y name, it contains the number of model factors, the PRESS, RMSEP,

7 Regression Methods: Factor Based Regression

SEP, the bias between these two measures, plus three quantities which characterize a plot of predicted value against reference value: correlation coefficient, slope and intercept.

If the SEP in this object and the training set SEV are comparable, it is likely that training set samples and prediction target samples are drawn from the same population and that your model contains an appropriate number of factors. If the model produces biased predictions, this will be reflected in significant deviations from an intercept of zero and a slope of one. Figure 7.29 shows the Error object for a prediction target containing five dependent variables.

Figure 7.29
PLS Error Analysis

		0.583955				
		1	2	3	4	5
		Heptane	Isooctane	Toluene	Xylene	Decane
1	RMSEP	0.5840	0.3515	0.2934	0.1830	0.4828
2	SEP	0.5840	0.3515	0.2934	0.1830	0.4828
3	Bias	0.0000	0.0000	0.0000	0.0000	0.0000
4	PRESS	10.2301	3.7069	2.5828	1.0041	6.9919
5	r	0.9956	0.9983	0.9989	0.9986	0.9987
6	Factors	7.0000	6.0000	6.0000	6.0000	7.0000
7	Slope	0.2745	0.3746	0.3565	0.3314	0.2313
8	Intercept	0.1529	0.0696	0.0843	0.0256	0.0363

Y Predictions

The predicted values for each dependent variable for each sample appear here in table format. Figure 7.30 which provides a side-by-side view of PLS and PCR predictions. Note that the number of model factors for each Y is embedded in the column title.

Figure 7.30
Comparing PLS and
PCR Y Predictions

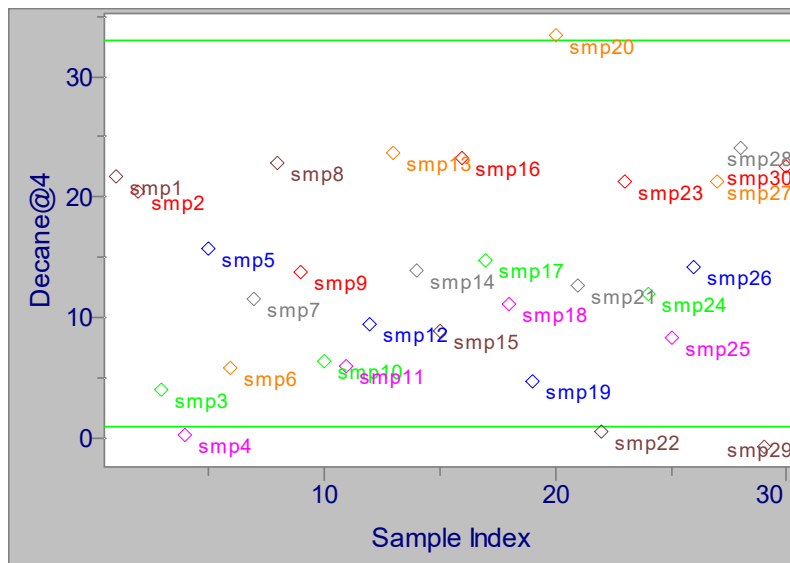
		10.140871				
		1	2	3	4	5
		Heptane@7	Isooctane@6	Toluene@6	Xylene@6	Decane@7
1	smp1	10.1409	9.4261	47.3871	13.4337	19.4966
2	smp2	13.1238	22.7652	30.1721	10.7147	23.2282
3	smp3	21.9649	26.3818	37.5142	12.9127	1.0243
4	smp4	11.2936	28.0159	44.3176	13.1421	3.4595
5	smp5	13.9234	22.1472	36.9719	6.9739	20.0452
6	smp6	12.9306	26.1429	50.0297	5.0332	5.9701

		10.635137				
		1	2	3	4	5
		Heptane@8	Isooctane@7	Toluene@6	Xylene@6	Decane@7
1	smp1	10.6351	9.3422	47.6633	13.2796	19.3551
2	smp2	13.5038	22.9360	30.8619	10.2666	23.3626
3	smp3	21.6771	26.2466	37.1534	13.0453	0.9127
4	smp4	11.0507	28.2847	44.6690	13.0173	3.8122
5	smp5	13.9993	22.1824	37.1495	6.8665	20.0767
6	smp6	12.5331	26.2301	50.0544	5.0347	6.3274
7	smp7	18.9770	29.0818	30.7088	4.9728	15.9493

A 2D plot of Y Predictions shows threshold lines which indicate bounds on the training set reference values for each Y (see Figure 7.31). This feature draws attention to instanc-

es of model extrapolation. Predictions lying inside the training set range are interpolated values while those lying outside the range are extrapolated. Model extrapolation is to be avoided.

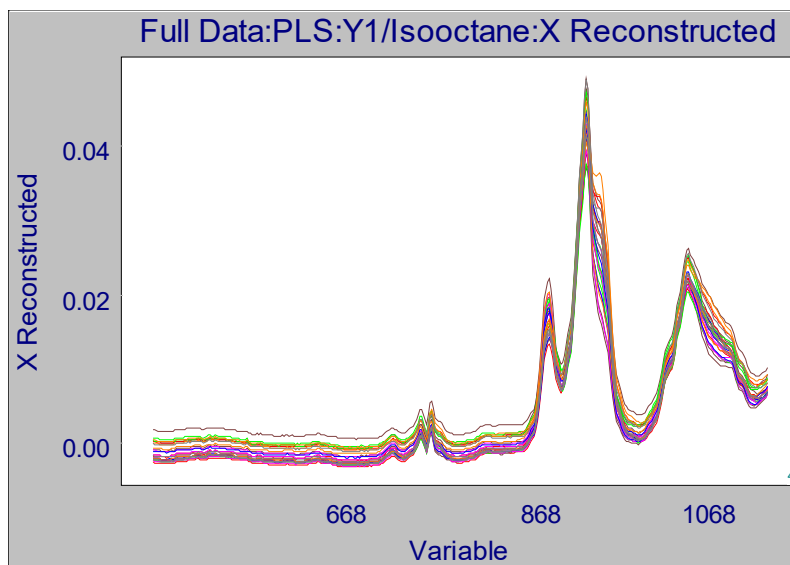
Figure 7.31
Scatter plot of Y
Predictions showing
Y range in model



X Reconstructed

This object is described in “X Reconstructed” on page 5-37.

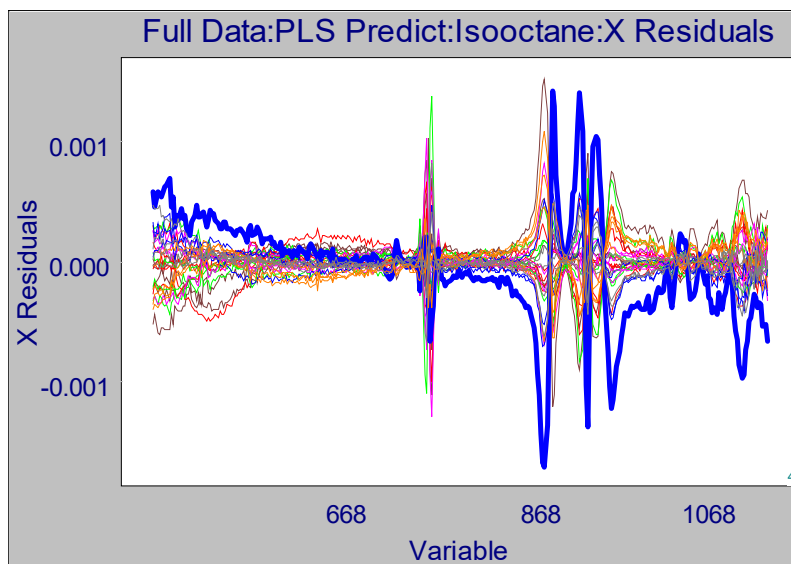
Figure 7.32
An X Reconstructed
object



X Residuals

The X residuals object produced during prediction is identical to the modeling object of the same name described on page 7-26. The figure shown below contains a line plot of the X Residuals from a small prediction set. Note that one of the samples (the highlighted blue trace) fits the model rather poorly, that is, it has large X residuals. Not surprisingly, this sample will present large outlier diagnostics (see “PLS Prediction Outlier Diagnostics” on page 7-36).

Figure 7.33
PLS Prediction X
Residuals

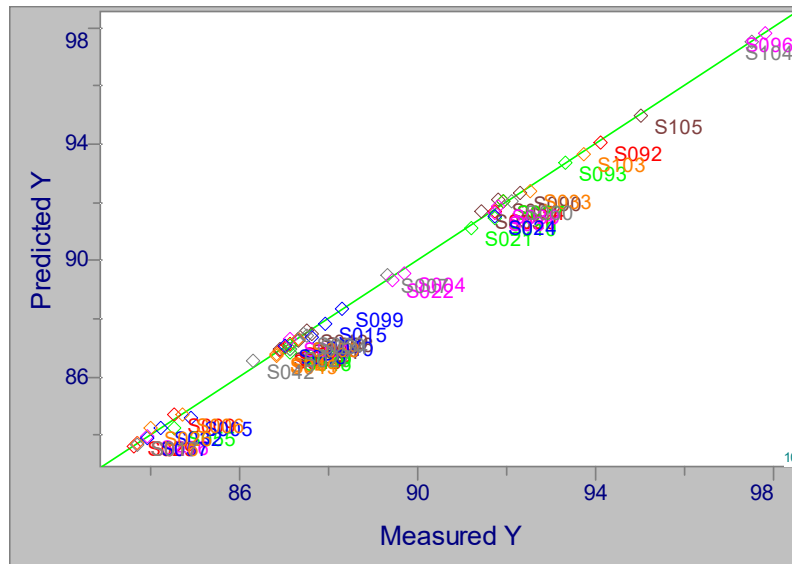


When attempting to determine the quality of any prediction, it is wise to compare the prediction X residuals to those associated with the training set to get an idea of a reasonable magnitude. Large X residuals during prediction imply that a sample does not belong to the same population as the training set. Variables having the largest residuals can indicate regions where the poorly predicting sample differs most from the training set samples. For example, in spectroscopic applications, if interferences are present in the prediction targets but absent from the training set, prediction X residuals will be large at variables where the interference makes a significant contribution to the signal. For more, see “Contributions” on page 5-40).

Y Fit

If the prediction target contains data for any dependent variables having the same name as the dependent variables in the training set, the Y Fit object is analogous to the object described in “Y Fit” on page 7-20. By default, when this object is dropped, it shows a plot of measured vs. predicted.

Figure 7.34
Prediction Y Fit
object



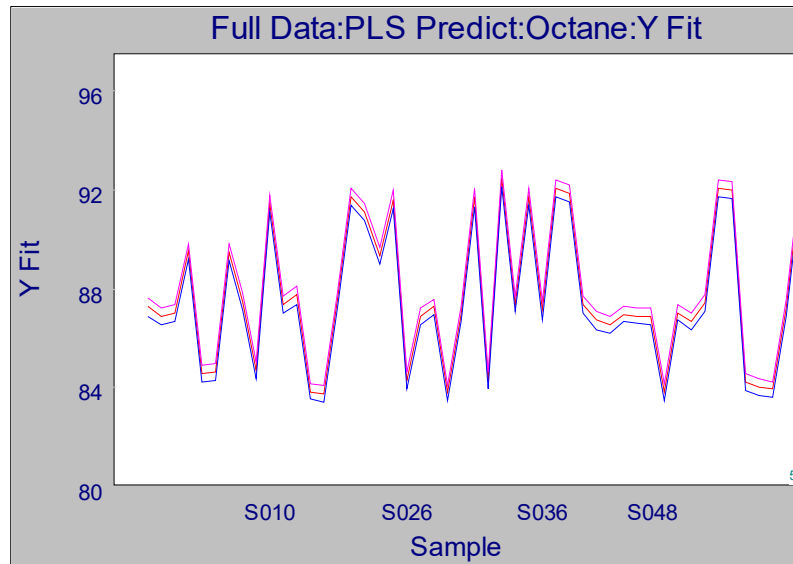
However, this object also contains the prediction confidence limits; see “[Prediction Confidence Limits](#)” on page 7-12. Switch to the table view to see all columns.

Figure 7.35
Prediction Y Fit
values and
confidence limits

Full Data:PLS Predict:Octane:Y Fit						
1,1		87.300003				
		1	2	3	4	5
		Measured Y	Predicted	Residual Y	Upper Limi	Lower Limi
1	S001	87.3000	87.2577	0.0423	87.5752	86.9402
2	S002	87.0000	87.0064	-0.0064	87.3189	86.6940
3	S003	87.1000	87.0666	0.0334	87.3724	86.7607
4	S004	89.7000	89.5339	0.1661	89.8376	89.2303
5	S005	84.9000	84.6062	0.2938	84.9202	84.2921
6	S006	84.7000	84.7141	-0.0141	85.0259	84.4022
7	S007	89.3000	89.5080	-0.2080	89.8124	89.2035
8	S008	87.6000	87.4806	0.1194	87.7876	87.1735
9	S009	84.5000	84.7393	-0.2393	85.0559	84.4227
10	S010	91.7000	91.4596	0.2404	91.7722	91.1469

If the prediction target does not contain dependent variable data related to the model, the Y Fit object contains only the predicted values and confidence limits. In this situation, the default view shows these values in a line plot in which the predicted values and their error bounds are overlaid.

Figure 7.36
Prediction Y Fit error
bounds

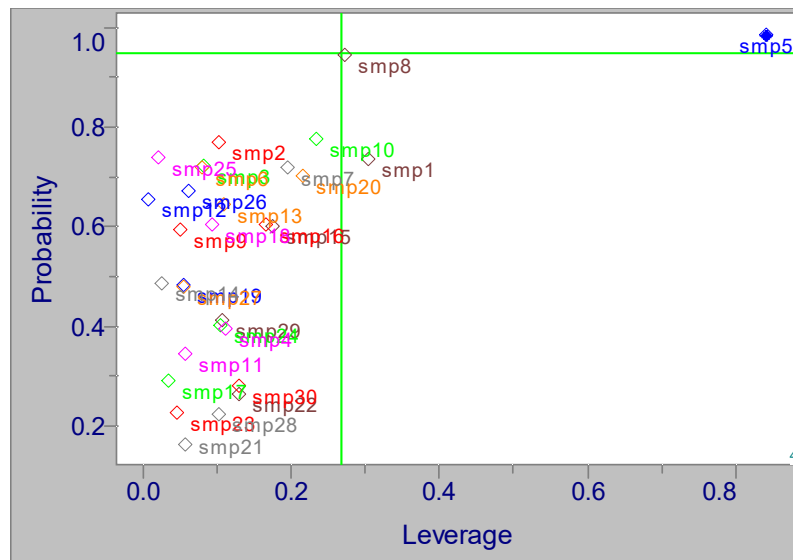


Outlier Diagnostics

To identify unknowns which differ significantly from the training set samples, an Outlier Diagnostic object is produced during prediction. It contains the Leverage, and Mahalanobis distance and Probability. The first quantity is described in “Leverage” on page 7-11 while the last two are discussed in “Mahalanobis Distance in Prediction” on page 5-31 and “Probability” on page 5-25, respectively.

The following figure shows the Outlier diagnostics corresponding to the data in Figure 7.33 above. The sample with the large X Residual, also highlighted below, is indeed an outlier sample, as indicated by large probability and leverage values.

Figure 7.37
PLS Prediction
Outlier Diagnostics

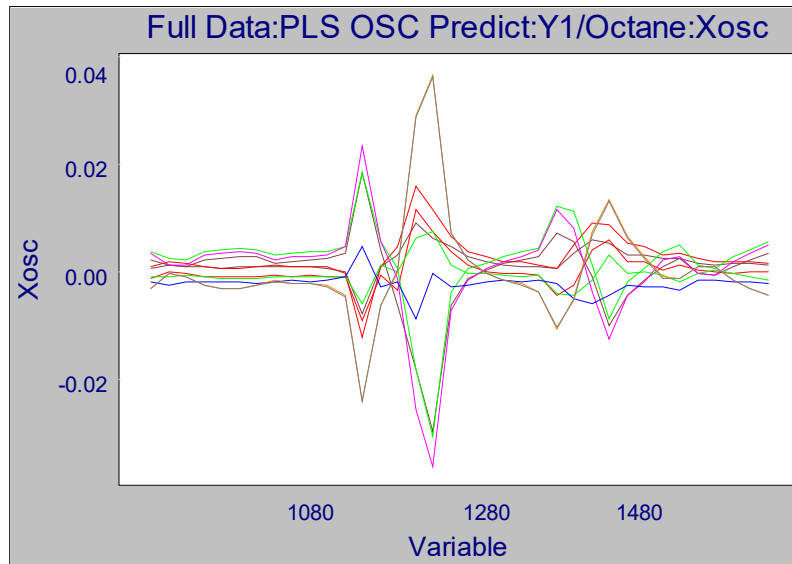


OSC Results

If OSC is enabled, when prediction is run, the data are orthogonalized before the algorithm; refer to equation 7.36 - equation 7.38. The orthogonalized X block is computed

and should be surveyed to look for important features particularly if they differ from the original X block.

Figure 7.38
Orthogonalized prediction data



The prediction quality can also be evaluated by contrasting the Error Analysis object with and without OSC.

Figure 7.39
Error analysis of PLS prediction without and with OSC

1,1		1	2	1,1		1	2
		Octane				Octane	
1	SEP	0.2343		1	SEP	0.2149	
2	PRESS	0.5488		2	PRESS	0.4616	
3	r	0.9969		3	r	0.9974	
4	Factors	6.0000		4	Factors	5.0000	
5	Slope	0.9872		5	Slope	1.0005	
6	Intercept	1.1833		6	Intercept	0.0080	
7	ModelESS	0.0005		7	ModelESS	0.0003	
8				8			
9				9			
10				10			
11				11			
12				12			
13				13			
14				14			
15				15			

Note that in this case, the errors were hardly reduced, but the bias, reflected in the slope and intercept, were much improved.

PLS for Classification

Although originally designed as a regression algorithm that correlated information in the X block to dependent variables, PLS has recently demonstrated some success in pattern recognition applications. In these scenarios, the Y variable is binary, with ones for samples that belong to the category and zeroes for non members. The approach is called PLS-DA to evoke its similarity to Discriminant Analysis. Pirouette's implementation automatically generates an implicit Y block from an existing class variable. The restrictions on this class variable are discussed in "Using Class Variables in Algorithms" on page 13-19. PLS-DA is based on the same mathematics as PLS. However, additional objects are computed, during both modeling and prediction; these are described in the next sections.

RUNNING PLS-DA

The parameters available when running PLS-DA are described in "PLS-DA Options" on page 16-27. Optimizing a PLS-DA model is similar to PLS and PCR. If you can afford to do so, run cross validation and look at the SEV results to help choose the optimal number of factors. Investigate the loadings and the regression vector to optimize the factor selection setting and to see if there are any regions in the X block that are not useful in the model. Look for unusual samples in the Outlier Diagnostic object and, if found, use the X Residuals and Contributions to understand what makes the samples different.

Class Predicted

PLS-DA classifications, like those in SIMCA, can result in a sample matching no model category, a single model category, or more than one model category. Accordingly the Class Predicted object embodies these three outcomes.

Figure 7.40
PLS-DA Class
Predicted object

		1	2	3	4	5	6
		Best	NextBest				
36	STAR2100	3	0				
37	STAR2100	3	0				
38	R01588	3	0				
39	R00955	0	0				
40	R03428	3	0				
41	STA40-140	4	0				
42	STA40-505	4	0				
43	STARGC4	4	0				
44	STARLB13	4	0				
45	STARKB32	4	0				
46	STARR039	4	0				
47	STARR034	4	0				

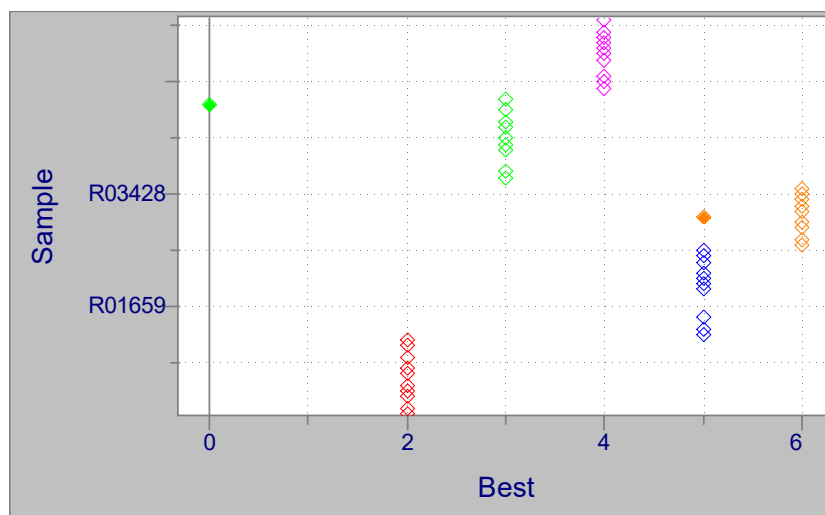
Like SIMCA, PLS-DA uses a special category name to designate no match: 0. The figure above shows that at least one sample was not classified into any category. It is possible, though rare, that during PLS-DA calibration a sample could be categorized into more

than one category. Tie-breaking criteria for this situation are discussed later; see “Classification Tie-breaker” on page 7-43.

Displaying the results as a plot can be helpful, particularly when the class variable used for the model is activated.

- Convert from Table to 2D view
- With the Selector button, put Sample Index on the Y-axis

Figure 7.41
PLS-DA Class
Predicted plot



One sample does not match any model category while, based on color consistency, we can see that another sample appears to be predicted into an incorrect category.

Misclassifications

A summary of the classification success of the PLS-DA model is presented in the Misclassification object (sometimes called a Confusion Matrix). The predictions for each category are summed and placed in a table.

Figure 7.42
PLS-DA
Misclassifications

		1	2	3	4	5	6
		Pred2@6	Pred3@6	Pred4@6	Pred5@6	Pred6@6	No match
1	Actual2	10	0	0	0	0	0
2	Actual3	0	9	0	0	0	1
3	Actual4	0	0	10	0	0	0
4	Actual5	0	0	0	10	0	0
5	Actual6	0	0	0	1	9	0
6							

If all samples predict into their pre-defined categories, values along the matrix diagonal represent the size for each class. Non-zero off diagonal values indicate classification errors. As shown in the previous figure, one sample did not match any model category,

while another sample was predicted in to an incorrect category. Note also that the column labels show the number of factors in each class' PLS model.

MAKING A PLS-DA PREDICTION

When ready to perform a PLS-DA prediction, use the same procedure as for PLS and PCR.

- Choose Process > Predict (or press Ctrl-Y)
- Select the Exclusion Set(s) on which to predict
- Select the Model(s) to use

An example of this process was already shown (see Figure 7.28, on page 7-31). When the prediction is complete, open the appropriate folder in the Object Manager to see the results.

Y Predictions

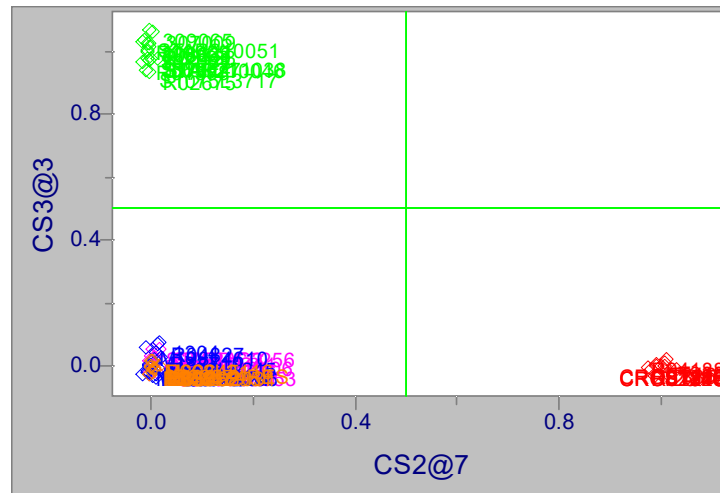
The default view of the Y Predictions object is that of a table; it contains a column for every category in the training set.

Figure 7.43
Y Predictions table;
highlighted cells
indicate
classifications

13,5		0.666797				
		1	2	3	4	5
		CS2@4	CS3@5	CS4@5	CS5@6	CS6@6
1	3A	0.002950	1.040636	-0.022614	-0.175797	0.168721
2	3B	0.004355	0.975642	-0.000407	0.077607	-0.070614
3	3C	0.010630	1.046480	-0.104705	0.238617	-0.232624
4	3D	-0.017225	1.076194	-0.068666	-0.220724	0.179127
5	4A	0.035857	-0.014417	1.112179	-0.155522	0.132059
6	4B	0.017961	0.016373	1.060481	-0.016368	0.010178
7	4C	0.007708	0.037197	0.946008	-0.000612	-0.011040
8	4D	0.000090	0.053666	0.936287	0.034860	-0.044799
9	5A	0.007697	-0.059496	0.072064	0.533240	0.491210
10	5B	0.004776	-0.084228	0.123117	0.844932	0.176972
11	5C	0.013018	-0.036039	0.042265	0.700254	0.287677
12	5D	-0.017044	-0.088628	0.060559	0.656763	0.345616

Because PLS-DA's implicit Y variables are set to one for the samples in the category and to zero otherwise, the predicted Y value should be close to one for samples truly belonging to the class and close to zero if not. In the preceding figure, samples that are clearly members of categories 3, 4, and 5, respectively, are shaded to differentiate them from the other predicted values. When this object is viewed as a scatter plot, reference lines appear.

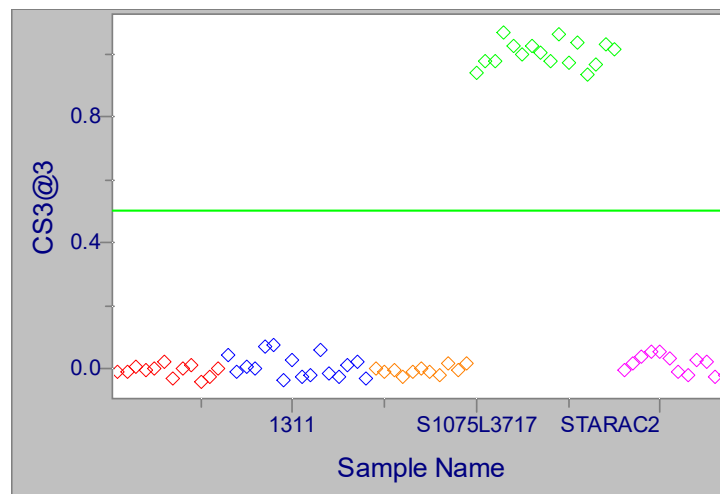
Figure 7.44
Y Predictions plot



In the PLS Y Predictions plot, the reference lines indicate the range of Y values in the model. In the case of PLS-DA, however, the reference lines indicate the decision criterion for class membership: only samples with Y values greater than 0.5 are called category members. Thus, in the plot, the red samples are all above the membership criterion for CS2, that is, class 2 which is plotted on the X axis while the green samples fall above the decision criterion of class 3, on the Y axis.

A 2D scatter plot with sample index on the x axis shows this more clearly.

Figure 7.45
Y Predicted plot of one category



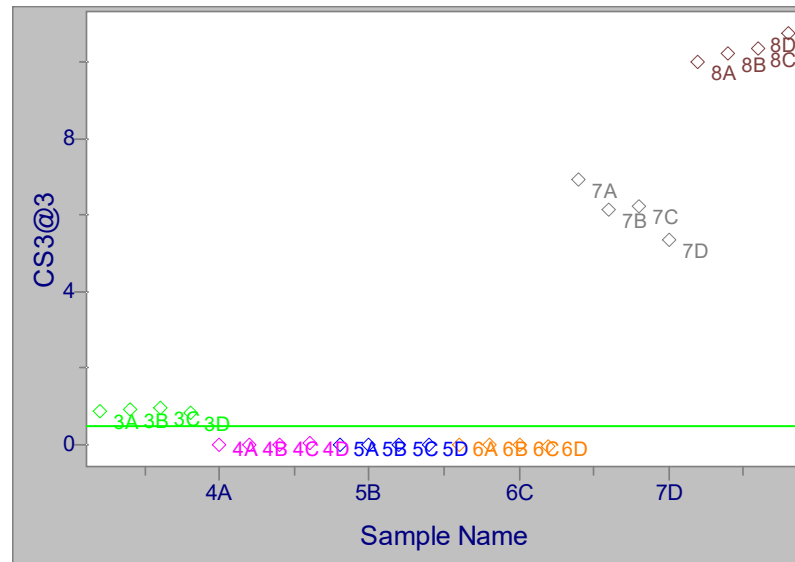
Thus, the green samples are the only ones classified into category 3.

Class Predicted

When performing predictions, we have to consider another case not possible during PLS-DA modeling: prediction samples might belong to categories not represented in the training set. The following figure is from another prediction in which some samples belong to training set categories but others do not.

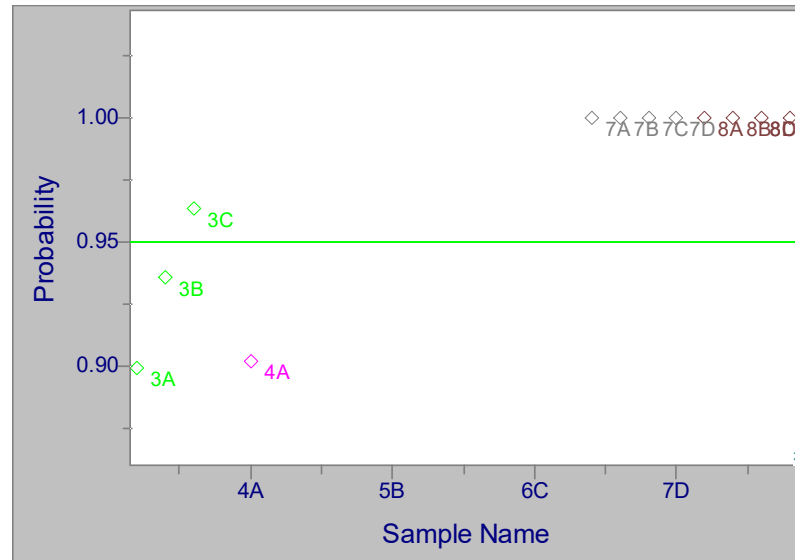
7 Regression Methods: PLS for Classification

Figure 7.46
Y Predicted plot with
unmodeled samples



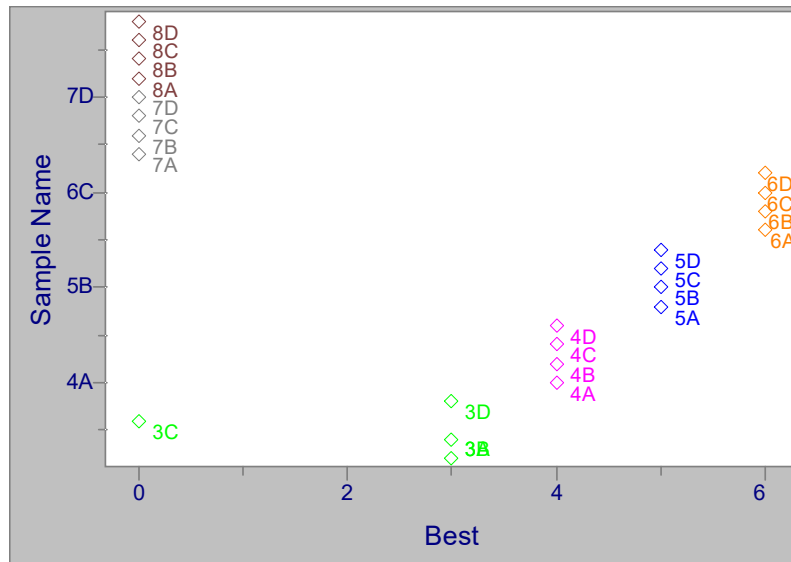
Clearly, some of the samples have predicted values considerably larger than 0.5. Pirouette uses the sample's Probability value to qualify the result (see "Probability" on page 7-48). Thus, any sample whose probability is greater than the Regression Probability setting in the Prediction Preferences dialog ("Prediction" on page 16-43), will be considered not classified, even if it exceeds the Y prediction value of 0.5. In this way, spurious classifications are avoided.

Figure 7.47
Prediction
Probabilities plot



Not only were the unmodeled samples disqualified, as shown in the previous plot, but one sample of a modeled category is also disqualified. These results are summarized by the Class Predicted object which considers both the Y Predicted and Probability results in the final category decision.

Figure 7.48
Class Predicted
object in prediction



Classification Tie-breaker

As with SIMCA, it is also possible for a sample to be classified by PLS-DA into two (or more) categories. This can occur when categories are very similar with overlapping factor spaces. In this situation, Pirouette applies a tie-breaker to favor one category: the sample is considered to better fit that class with the Y prediction closest to 1.

Note: This is an updated decision from version 4.0 which used the lowest probability for the tie-breaker.

Misclassification Matrix

As in the SIMCA result, the Misclassification Matrix object for prediction contains an extra row to summarize the results for unmodeled objects.

Figure 7.49
PLS-DA prediction
Misclassification
Matrix

		8.000000					
		1	2	3	4	5	6
		Pred2@6	Pred3@4	Pred4@6	Pred5@6	Pred6@6	No match
1	Actual2	0	0	0	0	0	0
2	Actual3	0	2	0	0	0	2
3	Actual4	0	0	4	0	0	0
4	Actual5	0	0	0	4	0	0
5	Actual6	0	0	0	0	4	0
6	Unmodelle	0	0	0	0	0	8
7							

Several outcomes are demonstrated in this table.

- No samples in class 2 were in the prediction set

- Two samples in class 3 were not classified into any category
- Eight samples in the prediction set were not in categories represented in the training set
- And, none of the 8 unmodeled samples were classified

Note: *The Misclassification Object will be computed only if there is a Class variable in the prediction set which has the same name as the Active Class used during the PLS-DA modeling phase. The name match is case sensitive. At least one sample must match a modeled category in order for the Misclassification object to be produced.*

Classical Least Squares

Although CLS is not strictly limited to applications which follow Beer's Law, it is most often mentioned in this context. For this reason the following discussion is developed in terms of Beer's Law and parameters associated with quantitative spectrophotometry.

MATHEMATICAL BACKGROUND

Consider the case where p is the number of components (*i.e.*, analytes) to be determined, n is the number of samples and m is the number of wavelengths at which absorbances are measured for each sample. \mathbf{X} is the n by m matrix of absorbances of the samples, \mathbf{Y} is the n by p matrix containing the concentration-pathlength product of the p components in the n samples, and \mathbf{K} is a p by m matrix. Each row of \mathbf{K} corresponds to the spectrum of one of the p analytes at unit concentration and unit pathlength. Each row of \mathbf{X} is the spectrum of one of the n samples. Using these definitions, the matrix formulation of Beer's Law is:

$$\mathbf{X} = \mathbf{Y}\mathbf{K} \quad [7.39]$$

This relationship embodies the additivity of Beer's Law. Any sample's spectrum is assumed to be the sum of the spectra of each of the p components in the sample. The absorbance spectrum due to any one of the p analytes is the product of its concentration and its pure component spectrum found in \mathbf{K} .

Note: *The extent to which Beer's Law holds **and** the additivity assumption is met influences CLS model quality. Moreover, if a component is present in samples but its pure component spectra is unavailable, CLS will perform poorly.*

The composition of new samples (*i.e.*, unknowns) can be determined immediately from their spectra and \mathbf{K} . Designating the spectrum of a single new sample as \mathbf{x}_{new} , [equation 7.39](#) can be rearranged to give a vector of estimated concentrations for this sample:

$$\hat{\mathbf{y}}_{\text{new}} = \mathbf{x}_{\text{new}}\boldsymbol{\beta} = \mathbf{x}_{\text{new}}\mathbf{K}^T(\mathbf{K}\mathbf{K}^T)^{-1} \quad [7.40]$$

where $\boldsymbol{\beta}$ is a **matrix** containing p column vectors, one for each component. The individual regression vectors are thus the columns of the pseudo-inverse of \mathbf{K} , symbolized by \mathbf{K}^\dagger :

$$\boldsymbol{\beta} = \mathbf{K}^T(\mathbf{K}\mathbf{K}^T)^{-1} = \mathbf{K}^\dagger \quad [7.41]$$

To account for non-zero spectral baselines, additional rows are added to \mathbf{K} , the number and content determining the degree of polynomial baseline fit for new samples. When r rows are added containing the independent variable indices raised to the $r-1$ power, a baseline of order $r-1$ is fit to each sample spectrum. Thus, to fit a constant baseline to each sample, a row of ones is added. To fit a linear baseline, an additional row is added containing 1, 2, 3, ... m . Every row added to \mathbf{K} produces a corresponding row in \mathbf{Y} . In the case of a linear baseline, the extra rows in \mathbf{Y} hold its intercept and slope.

Note: Failure to properly account for the baseline leads to poor quality predictions in CLS. This accounting takes place during the second inversion, when the pseudo-inverse of \mathbf{K} is computed.

Direct and Indirect CLS

If the \mathbf{K} matrix is already available, that is, the analytes have been specified and their pure component spectra taken from a library or measured by the analyst using pure analyte, the approach is called Direct CLS¹³. Its modeling (*i.e.*, calibration) phase is trivial, requiring no standards, only the computation of the pseudo-inverse of \mathbf{K} . However, Direct CLS may produce poor results in the prediction phase due to deviations from Beer's Law at high concentrations. A spectrum of neat analyte may differ significantly from the pure component spectrum of that analyte in dilute solution, where individual absorbers cannot "see" each other and can thus act independently. In concentrated solutions absorber-absorber interactions increase and influence the resulting spectrum. In other words, Beer's Law is a limiting relation, followed most closely in dilute solution. Moreover, library spectra contain features which depend on the particulars of instrument used to acquire them: resolution, signal-to-noise ratio, detector, etc. For these reasons, Indirect CLS is emphasized in Pirouette. However, Direct CLS is possible; it requires supplying the p by p identity matrix as a \mathbf{Y} block.

In Indirect CLS, pure component spectra are estimated from n training set samples which contain various amounts of the p components. Thus, an estimate of \mathbf{K} is determined from \mathbf{X}_{ts} , the training set spectra, and \mathbf{Y}_{ts} , training set concentration matrix:

$$\hat{\mathbf{K}} = (\mathbf{Y}_{ts}^T \mathbf{Y}_{ts})^{-1} \mathbf{Y}_{ts}^T \mathbf{X}_{ts} \quad [7.42]$$

To guarantee that the inverse of $\mathbf{Y}^T \mathbf{Y}$ exists, certain requirements on the number and composition of training set mixtures must be met. There must be as many independent standards as pure component spectra to be estimated. Moreover, special designs for \mathbf{Y}_k are necessary if samples have concentrations which sum to a constant value (*e.g.*, 100% by weight); see Cornell¹⁴ for a thorough discussion of mixture designs. Of course, the composition of the training set samples should span the prediction concentration space.

To guarantee that the inverse of $\mathbf{K} \mathbf{K}^T$ exists, certain requirements on the number of independent variables must be met. There must be as many independent variables as the number of dependent variables to be estimated plus the baseline order plus one. Thus, to determine the concentration of one species with a linear baseline (of order = 1), data for at least three wavelengths must be available.

Note: To simplify notation below, pure component spectra are represented by \mathbf{K} , not $\hat{\mathbf{K}}$.

Thus, Indirect CLS consists of two matrix inversions. The first results in an estimate of \mathbf{K} , the pure component spectra, and the second results in an estimate of \mathbf{Y} , the concentrations of the unknowns. In both cases, it is possible to write expressions which quantify the uncertainty in these estimates. These expressions, giving the uncertainties in regression coefficients, follow from standard multiple linear regression theory¹⁵.

Uncertainty of Pure Component Spectra

The reconstructed mixture spectra are calculated from

$$\hat{\mathbf{X}} = \hat{\mathbf{Y}}\mathbf{K} = \mathbf{X}_{ts}\beta\mathbf{K} \quad [7.43]$$

The residual portion of the mixture spectra is then

$$\mathbf{E}_{ts} = \mathbf{X}_{ts} - \hat{\mathbf{X}} = \mathbf{X}_{ts} - \mathbf{X}_{ts}\beta\mathbf{K} \quad [7.44]$$

The uncertainty in the estimate of pure spectra, $\Delta\mathbf{K}$, is a function of both \mathbf{Y}_{ts} and \mathbf{E}_{ts} , the residuals from the mixture spectra used to estimate \mathbf{K} :

$$\Delta\mathbf{k}_j = (\text{diag}(\mathbf{Y}_{ts}^T\mathbf{Y}_{ts})^{-1} s_j^2)^{1/2} \quad [7.45]$$

where $\Delta\mathbf{k}_j$ is the j^{th} column of $\Delta\mathbf{K}$, and the mean square about the regression, s_j^2 , is computed for each variable as:

$$s_j^2 = \frac{\mathbf{e}_j^T \mathbf{e}_j}{n - p} \quad [7.46]$$

where \mathbf{e}_j is the j^{th} column of \mathbf{E}_{ts} .

Uncertainty in the estimate of \mathbf{K} is determined by two factors: the experimental design of the training set mixture compositions (over which the user has some control) and spectral noise. Good experimental designs can minimize the magnitude of the elements of $(\mathbf{Y}^T\mathbf{Y})^{-1}$. Poor designs include almost redundant samples (*i.e.*, rows of \mathbf{Y} which are close to being linear combinations) with an attendant increase in the magnitude of the elements of $(\mathbf{Y}^T\mathbf{Y})^{-1}$. Obviously, as the magnitude of the random fluctuations in \mathbf{X}_{ts} increases due to more spectral noise, the magnitude of the elements of s_j must also increase.

Statistical Prediction Error (SPE)

Uncertainty in the y estimate produced by [equation 7.40](#) is given by:

$$\Delta y_i = (\text{diag}((\mathbf{K}\mathbf{K}^T)^{-1}) s_{\text{new}}^2)^{1/2} \quad [7.47]$$

where s_{new}^2 is the sample's residual variance defined in [equation 7.53](#) below based on \mathbf{e}_{new} . The uncertainty in a predicted y value is determined by two factors. First, the quality of the fit of the sample's spectrum affects s_{new} ; samples which are indeed mixtures of only the p components in the training set will have small residual variances. Second, the degree of spectral overlap in the Pures influences the magnitude of the elements of \mathbf{K} . As the overlap in the Pures increases, so does the diagonal of $(\mathbf{K}\mathbf{K}^T)^{-1}$.

Cross Validation

The idea of cross-validation is discussed in "[Model Validation](#)" on [page 5-19](#) where the emphasis is on choosing the number of factors for PCA, PLS, or PCR. In CLS, an anal-

ogous choice is the order of the baseline fit and the main motivation is to improve the estimates of model quality and make outliers easier to find. Cross-validation in CLS is quite speedy so you are encouraged to choose this option when running the algorithm. It differs slightly from that in PCA, PLS, and PCR. In addition to a vector of predicted concentrations generated for each left out sample, the X Residuals are also computed and stored. Thus, all quantities derived from the X Residuals differ from those produced when no validation is specified for a Run Configuration.

Y Residuals

For any sample with a known concentration, the prediction residual for any y is:

$$\hat{f} = y - \hat{y} \quad [7.48]$$

where y is the “true” value for the dependent variable. For a set of n samples, a Prediction Residual Error Sum of Squares (or PRESS) can be calculated for the y variable under consideration:

$$\text{PRESS} = \mathbf{f}^T \mathbf{f} \quad [7.49]$$

Related to the PRESS is the Standard Error of Prediction (SEP), which takes into account the number of samples and has the same units as the y variable:

$$\text{SEP} = \left(\frac{\text{PRESS}}{n} \right)^{1/2} \quad [7.50]$$

The most naive version of validation predicts on the training set samples. This type of SEP is termed a Standard Error of Calibration (SEC):

$$\text{SEC} = \left(\frac{\text{PRESS}}{n - p} \right)^{1/2} \quad [7.51]$$

When cross-validation is specified, the quantity is symbolized by SEV to distinguish it from the SEP which is produced by predictions of true test samples.

Outlier Diagnostics

It is always desirable to remove unrepresentative training set samples before building a model. For CLS, Sample Residual, F ratio, Probability and Mahalanobis distance are calculated to help pinpoint unusual samples.

Sample Residual

A sample’s residual variance follows directly from the residual matrix \mathbf{E}_{ts} . To make the notation less cumbersome, the subscript ts will be dropped. The i^{th} row of \mathbf{E} , a vector \mathbf{e}_i , is the difference between that sample’s original data and its estimate, $\hat{\mathbf{x}}_i$:

$$\mathbf{e}_i = \mathbf{x}_i - \hat{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x}_i \beta \quad [7.52]$$

A sample’s residual variance is then:

$$\hat{s}_i^2 = \frac{\mathbf{e}_i \mathbf{e}_i^T}{m - p} \quad [7.53]$$

In Pirouette, the square root of sample residual variance is called the *sample residual*:

$$\hat{s}_i = \left(\frac{\mathbf{e}_i \mathbf{e}_i^T}{m-p} \right)^{1/2} \quad [7.54]$$

A total variance can be calculated for the whole training set:

$$s_0^2 = \frac{1}{n-p} \sum_i^n \hat{s}_i^2 \quad [7.55]$$

F Ratio

If a particular sample residual is larger than s_0 , it is natural to wonder if the sample is an outlier, *i.e.*, it might not belong to the same population as the other samples in the training set. An F test is used to decide if two variances differ significantly, the appropriate ratio being:

$$F_i = \frac{\hat{s}_i^2}{s_0^2} \quad [7.56]$$

As F_i gets large, the likelihood increases that the sample is not drawn from the same population as the other training set samples.

If the left-hand side of [equation 7.56](#) is set equal to a critical value extracted from an F table (based on 1 and $n-p$ degrees of freedom and a user-specified probability), a critical sample residual can be determined by rearrangement:

$$s_{\text{crit}} = s_0 (F_{\text{crit}})^{1/2} \quad [7.57]$$

This then becomes a threshold for deciding whether a sample residual is “too large”. If a sample residual exceeds s_{crit} , that sample may be an outlier.

Probability

Another way to flag unusual samples is by determining the probability associated with the quantity in [equation 7.56](#) assuming an F distribution with 1 and $n-p$ degrees of freedom. As a sample’s probability approaches 1, the chance it is an outlier increases.

Mahalanobis Distance

For each sample and each dependent variable, a Mahalanobis distance is computed:

$$MD_i = (\hat{\mathbf{y}}_i - \bar{\mathbf{y}})^T \mathbf{S}^{-1} (\hat{\mathbf{y}}_i - \bar{\mathbf{y}}) \quad [7.58]$$

where \mathbf{S} is the covariance matrix of $\hat{\mathbf{y}}_i$, and $\bar{\mathbf{y}}$ is the mean predicted y . Assuming that Mahalanobis distance is normally distributed, a critical value MD_{crit} can be determined from a chi squared distribution with p degrees of freedom. If a sample’s Mahalanobis distance exceeds MD_{crit} , that sample may be an outlier.

RUNNING CLS

The options associated this algorithm are shown in the figure below. When it executes, many objects are computed which can help you find sample outliers and make decisions

about excluding variables. Each is described below along with ideas about how to examine them.

Figure 7.50
CLS Options

In addition to the computed objects, information necessary to make predictions for each dependent variable included in the training set is stored in memory as pieces of a regression model. A model can be used as soon as it has been created or it can be stored separately from the training set data and reloaded later to make predictions on future samples. A Pirouette CLS model is more than just a matrix of regression vectors. It also contains information about which variables were excluded and what transforms/preprocessing options were chosen so that future samples are treated in the same way.

Model building is an iterative process. You will seldom run a regression algorithm just once and immediately start making predictions. Instead you will spend much of your time optimizing your model, that is, finding the “best” set of samples, variables and baseline order.

Baseline Select

Each type of baseline is associated with an integer from 1 to 5 corresponding respectively to none, constant, linear, quadratic and cubic fit. The model error sum of squares (ESS), defined below, is computed for each baseline setting:

$$ESS = \sum_i^n \mathbf{e}_i \mathbf{e}_i^T \tag{7.59}$$

It is displayed on the y axis as a function of the five baseline integers on the x axis in the Baseline Select plot. Changing the diamond handle position triggers recomputation of all quantities dependent on the baseline fit. Plots of such quantities display in their lower right corner the integer corresponding to the current Baseline Select setting.

Errors

For each of the p components, the PRESS CAL and the SEC are computed. The correlation coefficient r_{Cal} for the predicted y vs. known y is also displayed. Performing cross-validation triggers computation of the validation analogs of these three quantities.

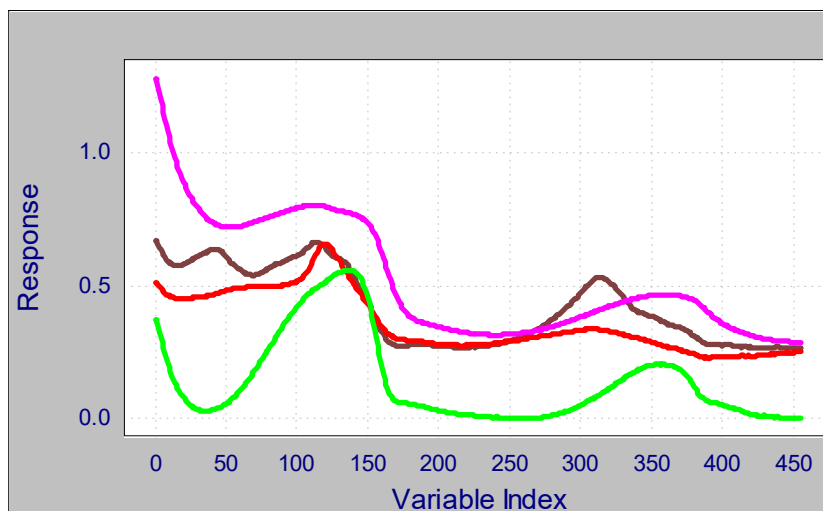
Figure 7.51
CLS Errors

		1	2	3	4
		UREA	CREATININE	NaCl	WATER
1	RMSEV	0.000962	0.000730	0.000523	0.001991
2	Press Val	0.000029	0.000017	0.000008	0.000123
3	rVal	0.994906	0.996075	0.998294	0.992536
4	RMSEC	0.000879	0.000659	0.000476	0.001835
5	Press Cal	0.000021	0.000012	0.000006	0.000091
6	rCal	0.996286	0.997226	0.998773	0.994476

Pures

The estimates of the pure component spectra given by [equation 7.42](#) are stored in the Pures object.

Figure 7.52
CLS Pures

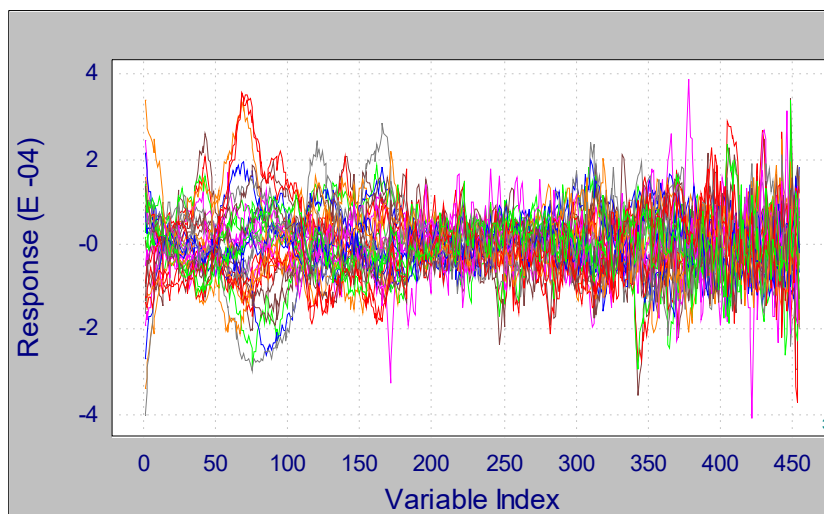


A line plot view of the Pures presents the amount of spectral overlap. Remember that highly overlapped spectra tend to produce poorer CLS models. You should examine the individual pures to confirm that each indeed resembles the spectrum of analyte. You may want to compare the estimated pures to library spectra of the analytes under investigation.

X Residuals

The X Residuals are the portion of the training set spectra not fit by the estimated Pures and the estimated Ys. Ideally, these residuals should have no structure but in practice, they often increase at wavelengths where the signal is large. When the magnitude of the residuals approaches a significant fraction of the signal being modeled, the model is inadequate.

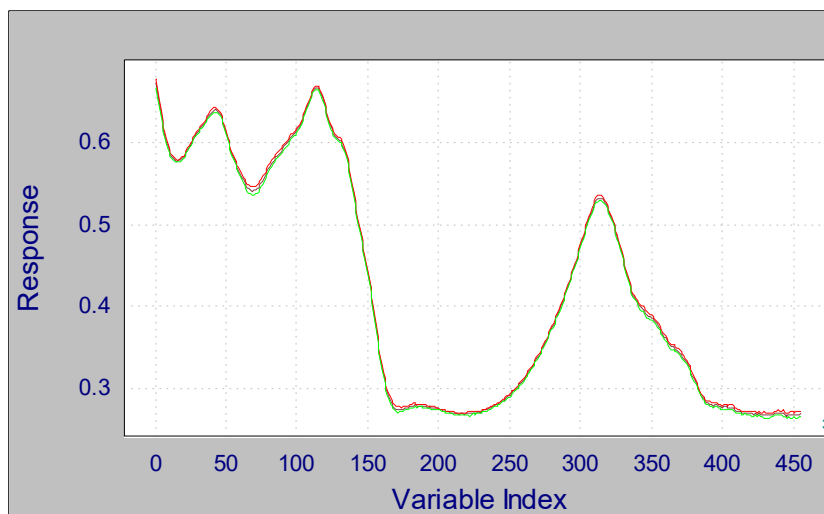
Figure 7.53
CLS X Residuals



Bounded Pure

Because the Pures are part of any subsequent prediction on new samples, errors in the Pures produce errors in predicted y values. The imprecision in the estimates of a Pure is given by [equation 7.45](#). A sort of confidence interval can be formed around each Pure by multiplying the Uncertainty, a standard deviation, by a constant based on the Probability Threshold set in the Run Configure dialog. For example, a Probability Threshold of 0.95 corresponds to a multiplier of 1.96. Thus, the lower bound is the Pure minus 1.96 times the Uncertainty and the upper bound, the Pure plus 1.96 times the Uncertainty.

Figure 7.54
CLS Bounded Pure

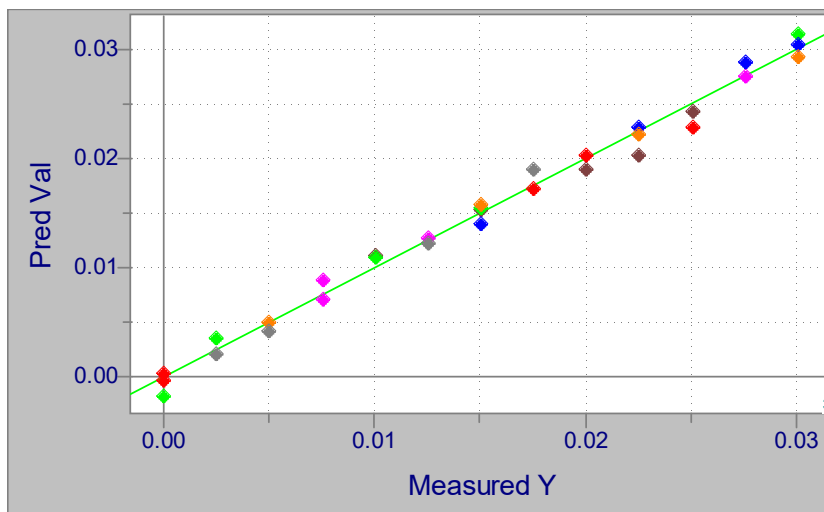


In the plot shown above, the bounds are practically superimposed on the Pure estimate itself, implying a very good fit. If the magnitude of the uncertainty is large compared to your estimate of the maximum allowed spectral noise, you may be able to decrease it by changing the experimental design of Y or by increasing the signal to noise ratio of the training set spectra. If the maximum magnitude of the uncertainty is on the order of the uncertainty in the instrument used to acquire the spectra, the design of the training set is acceptable.

Y Fit

For each component included in the training set, an object very similar to one of the same name computed during factor based regression is provided (see “Y Fit” on page 7-20). It differs only by including the SPE, described in “Statistical Prediction Error (SPE)” on page 7-46.

Figure 7.55
CLS Y Fit

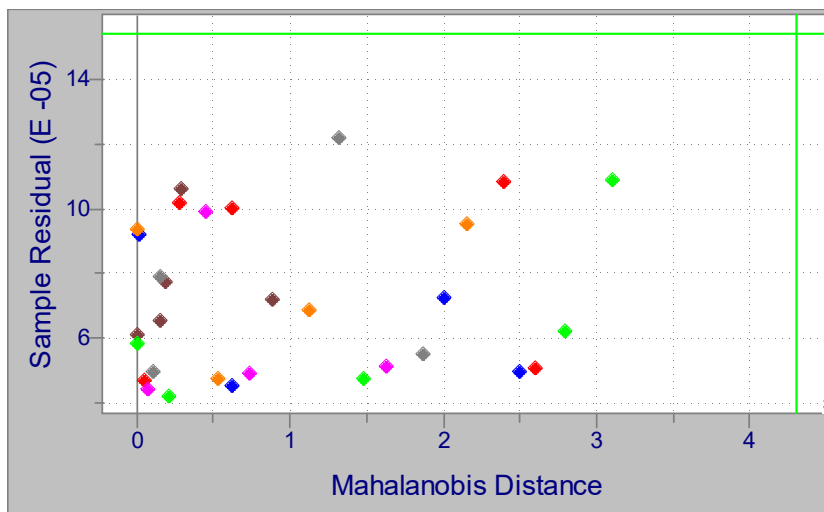


Recall that the SPE estimates the precision error in the predicted Ys. You should confirm that the SPE for each analyte is less than the maximum acceptable imprecision stipulated when the regression problem was defined. Training set samples with too large SPE values indicate an inadequate CLS model.

Outlier Diagnostics

This object contains the Sample Residual (defined by equation 7.54), Mahalanobis distance (defined by equation 7.58), F Ratio (defined by equation 7.56) and Probability for each training set sample.

Figure 7.56
CLS Outlier Diagnostics



Approximate threshold lines are provided to flag unusual samples. Samples lying well beyond one or more thresholds are potential outliers. Exclude them and rebuild the model, then decide if their inclusion is detrimental.

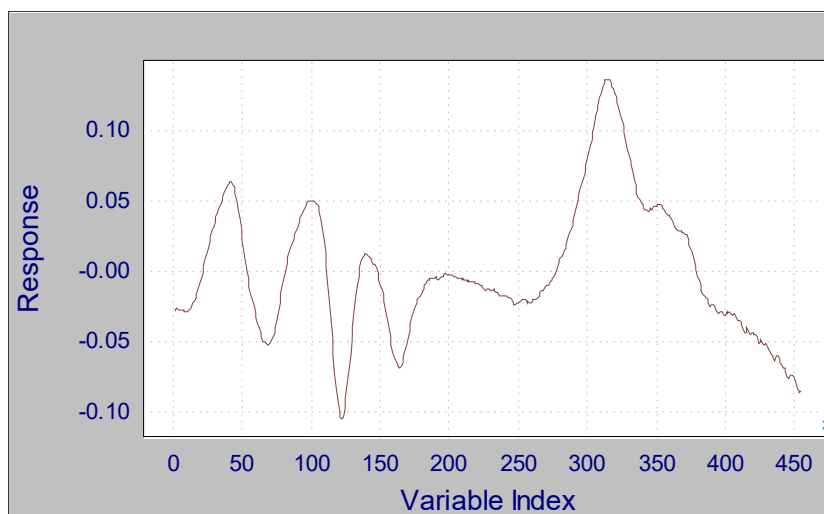
X Residuals

The X Residuals are the portion of the training set spectra not fit by the estimated Pures and the estimated Ys. Ideally, these residuals should have no structure but in practice, they often increase at wavelengths where the signal is large. When the magnitude of the residuals approaches a significant fraction of the signal being modeled, this indicates that the model is inadequate.

Regression Vector

For each component, the regression vector is also supplied, such as in the example shown below.

Figure 7.57 CLS Regression Vector



MAKING A CLS PREDICTION

Running CLS triggers the creation of a model. You can confirm this by going to Process/Predict after running the algorithm and noting the entry under Model. Making predictions requires a model and a target, *i.e.*, a data set with an X block containing one or more samples whose y values will be predicted. The target's X block must contain the same number of independent variables as the data set from which the model was created and no independent variables may be excluded. The target may or may not contain dependent and class variables.

Figure 7.28 shows the Configure Prediction dialog box. To get model information, highlight its entry as illustrated in that figure. To configure a prediction, highlight a model and an exclusion set and click on Add. You configure more than one prediction at a time by highlighting a different model name or exclusion set then clicking Add. Predictions are made when you click Run. The several objects available after CLS prediction is performed are described below.

Baseline Select

You can change the baseline type when predicting. This object contains the model error sum of squares and the standard errors for each Pure. If cross-validation was run, the SEV is shown, otherwise it's the SEC.

Y Predictions

Predicted values for all Ys included in the model are stored in this object.

Figure 7.58
CLS Predictions

		1	2	3	4
		UREA	CREATININE	NaCl	WATER
1	CT23ON01	0.020650	0.005605	0.009103	0.963841
2	CT23ON02	0.022656	0.002510	0.028445	0.946461
3	CT23ON03	-0.002560	0.026596	0.000839	0.973501
4	CT23ON04	0.025887	0.005110	0.010912	0.962518
5	CT23ON05	0.028998	0.022622	0.023235	0.925583
6	CT23ON06	0.003686	0.029701	0.026428	0.943593
7	CT23ON07	0.010914	0.025389	0.003090	0.962842
8	CT23ON08	0.017763	0.011457	0.018196	0.952200
9	CT23ON09	0.015825	0.020207	0.016056	0.950887
10	CT23ON10	0.008833	0.009915	-0.002059	0.987047
11	CT23ON11	0.005334	0.019183	0.004902	0.972570
12	CT23ON12	0.012376	0.013342	0.020640	0.955764

Error Analysis

This object is very similar to one of the same name computed for factor based regression, except that the number of model factors is replaced by the integer corresponding to the chosen baseline type. See “Error Analysis” on page 7-31 for a discussion of this object.

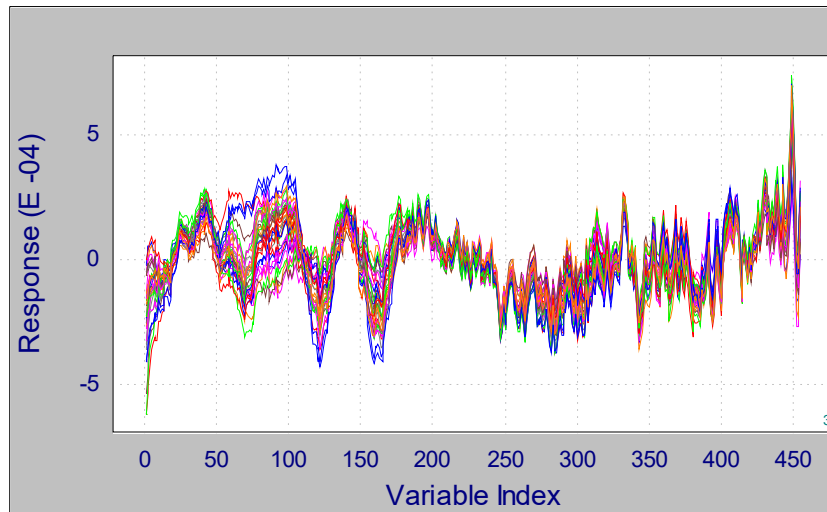
Figure 7.59
CLS Error Analysis

		1	2	3	4
		UREA	CREATININE	NaCl	WATER
1	RMSEP	0.001888	0.003543	0.002324	0.003198
2	SEP	0.000611	0.000990	0.000667	0.001793
3	Bias	0.001787	-0.003401	0.002226	-0.002649
4	PRESS	0.000111	0.000389	0.000167	0.000317
5	r	0.998021	0.993005	0.997260	0.994050
6	Slope	1.009536	1.008586	1.002676	1.006470
7	Intercept	-0.001932	0.003289	-0.002267	-0.003539
8	Baseline	3.000000	3.000000	3.000000	3.000000

X Residuals

The X Residuals produced during prediction are interpreted in much the same way as those found during the calibration; see “X Residuals” on page 7-50.

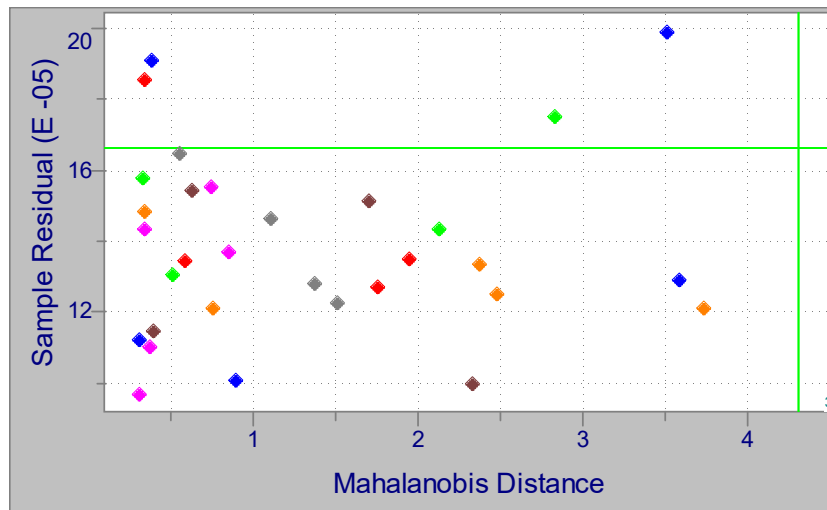
Figure 7.60
CLS Prediction X
Residuals



Outlier Diagnostics

The Outlier Diagnostics produced during prediction are interpreted in much the same way as those found during the calibration (see “Outlier Diagnostics” on page 7-36). An example of this object is shown below.

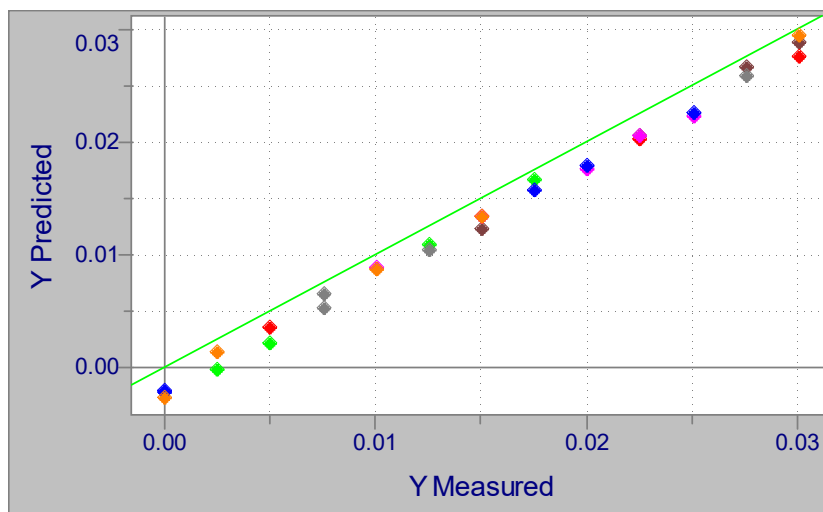
Figure 7.61
CLS Prediction
Outlier Diagnostics



Y Fit

If the sample being predicted is accompanied by a known y value whose name corresponds to a y value name in the training set, this object is identical to that produced during modeling; see “Y Fit” on page 7-52. If no known value of y is available (or if the y value name does not match), only the predicted Y and SPE are displayed.

Figure 7.62
CLS Prediction Y Fit



Calibration Transfer

Regression models, when saved to a file, are portable and can be shared with other Pirouette users. However, differences among instruments may be great enough to make predictions using shared models unreliable. Transfer of calibration approaches may allow such models to be used with little or no loss of reliability. For more background on this topic, see “Calibration Transfer” in Chapter 4.

To transfer a calibration during regression prediction, you must use a model derived from an algorithm configured to Enable Calibration Transfer. The check box for this option appears in Figure 16.27, on page 16-25. Along with the profiles (that is, the x block) to be adjusted, you must also supply a class variable and y variables and choose the calibration transfer type. The contents of and constraints on the variables are described below.

REQUIRED VARIABLES

The Adjust Mask class variable determines which prediction samples are candidate transfer samples; it must contain only 1s and 0s and must contain at least one 1. Samples flagged with a 0 are excluded from calibration transfer calculations; samples flagged with a 1 may be involved in the calibration transfer calculations, depending on other conditions. The name of the Adjust Mask variable is specified in Prediction Preferences dialog (see “Prediction” on page 10-19).

It is mandatory that the prediction set include a y block with exactly the same y variable names as in the model. If every y variable name in the model is not present in the prediction set, a calibration cannot be transferred and the prediction aborts. If the y names do match, then for each candidate transfer sample, the values of each y variable are examined and compared to the model values. If *these* values do not match, again the prediction aborts.

To show which training set samples are stored in the model and thus indicate possible matching y settings, an object called Ranked Transfer Samples is computed whenever a regression algorithm is run with Enable Calibration Transfer checked. It lists the sample names in optimal order and their y block values. Ideally, as many transfer samples as pos-

sible, starting from the top of the Ranked Transfer list, should be included in the prediction set to assure good reliability in the calibration transfer.

Figure 7.63
Ranked Transfer
Samples for PLS

		1	2
		Octane	
1	S090	92.3000	
2	S105	95.0000	
3	S057	83.9000	
4	S104	97.5000	
5	S024	91.7000	
6	S099	88.3000	
7	S015	87.9000	
8	S092	94.1000	
9	S103	93.7000	

CALIBRATION TRANSFER OPTIONS

The last three items in the Regression group of the Prediction Preferences dialog shown below apply to calibration transfer. There you specify the name of the Mask Variable, the transfer type and Window Size, which is applicable only for Piecewise transfers. For background on the different types, see “Calibration Transfer” on page 4-33.

Figure 7.64
Regression
Prediction
Parameters

Regression

Probability: 0.950000 (0 - 1)

Mask Variable: Mask

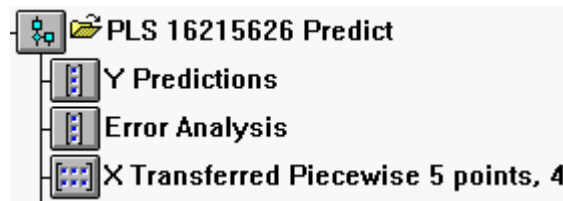
Calibration Transfer Type: None

Window Size: 1

X TRANSFERRED

When calibration transfer is applied during regression prediction, the results include an additional object, whose name contains the transfer type and the number of transfer samples. For example, the figure below shows that a 5 point Piecewise transfer was applied using 4 transfer samples. This object contains the x block of the prediction samples after adjustment. You should always compare this object to the original transformed prediction profiles. Similarly, it is wise to compare the predicted y values with and without calibration transfer.

Figure 7.65
X Transferred



Locally Weighted Regression

Not all training sets for regression analysis are as homogeneous as we might want. There are several strategies for accommodating scenarios in which the distribution of samples varies. One of these is termed Locally Weighted Regression (LWR) in which a regression model is developed on a reduced set of samples selected by proximity, in data space, to the prediction sample.¹⁶

LWR MODELING

Typically, in LWR the user decides how many samples to include in the local model (number of neighbors), how many factors would be optimal for that model, and whether the modeling is done with PLS or PCR.

Figure 7.66
LWR options

Locally Weighted Regression	
Preprocessing:	Mean-center
Maximum Factors:	5 (0-0)
# Neighbors:	25
Regression Alg:	PLS

Ideally, an optimization would be run to determine the first two parameters (contact [Infometrix](#) if this is a service you wish to contract), but may be unnecessary if one already has experience with the data. No further interaction with the algorithm is necessary; a model will be saved after LWR is run, and this model will contain all information necessary to perform a LWR prediction.

Note: A LWR model contains information for only a single Y variable. If your data contains more than one Y, include only the one you wish to model.

LWR PREDICTION

During a LWR prediction, a custom model is made for each prediction sample. First, the N samples (where N is defined in the modeling settings) nearest to the prediction sample, in data space, are found. Then a regression model (either PLS or PCR, from the settings) is created. Finally, a prediction is made on the sample from this model, using k factors, where k is also defined in the settings.

Results generated during a LWR prediction are similar to those for a regular regression prediction. These include:

- Y Predictions
- Error Analysis (only if the prediction data includes a Y variable with the same name, case sensitive, as that in the training data)
- Outlier Diagnostics
- YFit (if the Y name is absent, will only contain the predicted values and bounds)
- X Residuals
- X Reconstructed

References

1. Martens, H. and Næs, T.; *Multivariate Calibration*, (Chichester: John Wiley & Sons, 1989).
2. Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; and Vetterling, W.T.; *Numerical Recipes* (Cambridge: Cambridge University Press, 1986), pp. 52-64.
3. Wold, H.; "Partial Least Squares", in S. Katz and N.L. Johnson, Ed., *Encyclopedia of Statistic Sciences, Vol. 6* (Wiley: New York, 1985), pp. 581-591.
4. Manne, R.; "Analysis of Two Partial-Least-Squares Algorithms for Multivariate Calibration", *Chemom. Intell. Lab. Syst.* (1987) 2: 187-197.
5. Haaland, D.M. and Thomas, E.V.; "Partial Least-Squares Methods for Spectral Analysis. 1. Relation to Other Quantitative Calibration Methods and the Extraction of Qualitative Information", *Anal. Chem.*, (1988) 60:1193-1202.
6. Osten, D.W.; "Selection of Optimal Regression Models via Cross-Validation", *J. Chemometrics*, (1988) 2: 39.
7. Pell, R.J., Ramos, L.S. and Manne, R.; "The model space in Partial Least Squares regression", *J. Chemometrics* (2007) 21:165-172.
8. Gowen, A. A.; Downey, G.; Esquerre, C. and O'Donnell, C. P.; "Preventing overfitting in PLS calibration models of near-infrared (NIR) spectroscopy data using regression coefficients." *Journal of Chemometrics* (2011) 25(7): 375-381.
9. Weisberg, S.; *Applied Linear Regression* (New York: John Wiley & Sons, 1985), Chapter 5.
10. Cook, R.D. and Weisberg, S.; *Residuals and Influence in Regression* (New York: Chapman and Hall, 1982), Chapter 2.
11. Faber, N.M.; Schreutelkamp, F.H. and Vedder, H.W.; "Estimation of prediction uncertainty for a multivariate calibration model", *Spectroscopy Europe* (2004) 16:17-20.
ASTM E 1655 Standard Practices for Infrared Multivariate Quantitative Analysis, (ASTM International, West Conshohocken, PA).

12. Westerhuis, J.A., de Jong, S. and Smilde, A.K.; "Direct orthogonal signal correction", *Chemom. Intell. Lab. Syst.* (2001) 56: 13-25
13. Beebe, K.R.; Pell, R.J.; and Seasholtz, M.B.; *Chemometrics: A Practical Guide*, (New York: John Wiley & Sons, 1998).
14. Cornell, J.A.; *Experiments with Mixtures*, 2nd ed. (New York: Wiley, 1990).
15. Draper, N. and H. Smith; *Applied Regression Analysis*. 2nd Edition (New York: Wiley, 1981).
16. Centner, V. and D. L. Massart; "Optimization in locally weighted regression." *Analytical Chemistry* (1998) 70(19): 4206-4211.

Reading List

1. Beebe, K.R. and B.R. Kowalski; "An Introduction to Multivariate Calibration and Analysis", *Anal. Chem.* (1987) 59: 1007A-1017A.
2. Geladi, P. and Kowalski, B.R.; "Partial Least-Squares Regression: A Tutorial", *Anal. Chim. Acta* (1986) 185: 1-17.
3. Haaland, D.M.; "Classical Versus Inverse Least-Squares Methods in Quantitative Spectral Analyses", *Spectroscopy*, 2(6):56-57 (1987).
4. Lorber, A., Wangen, L.E. and Kowalski, B.R.; "A Theoretical Foundation for the PLS Algorithm", *J. Chemom.* (1987) 1: 19-31.
5. Martens, H. and Næs, T.; *Multivariate Calibration* (Chichester: John Wiley & Sons, 1989).
6. Massart, D.L.; Vandeginste, B.G.M.; Deming, S.N.; Michotte, Y.; and Kaufman, L.; *Chemometrics: a textbook*, (Amsterdam: Elsevier, 1988).
7. Næs, T.; Isaksson, T.; Fearn, T.; and Davies, T.; *Multivariate Calibration and Classification* (Chichester, NIR Publications, 2002).
8. Sharaf, M.A., Illman, D.L., and Kowalski, B.R.; *Chemometrics* (New York: John Wiley & Sons, 1986).
9. Thomas, E.V. and D.M. Haaland; "Comparison of Multivariate Calibration Methods for Quantitative Spectral Analysis", *Anal. Chem.* (1990) 62: 1091-1099.

Mixture Analysis

Contents

Introduction	8-1
Alternating Least Squares	8-3
Multivariate Curve Resolution	8-12
Reference	8-27

Chemists often encounter mixtures whose composition may not be completely characterized. These situations range from unresolved chromatographic peaks to reservoir oil samples having contributions from more than one source rock. Mixture analysis is the general term for techniques applied to mathematically separate the components in these types of mixtures.

Introduction

Consider a familiar problem scenario: a chromatographic peak is suspected to be impure, that is, it might contain two or more coeluting compounds. It would be desirable to determine the number of coelutes and express the observed peak profile as a sum of contributions from each. Chromatographers often refer to this approach as curve resolution or curve deconvolution.

Now consider a second problem scenario: environmental samples collected at various geographical locations are suspected to originate from one or more point sources. Again it would be desirable to determine the number of contributing sources and the relative amounts of each in the samples. This approach is often referred to as source apportionment.

If samples are characterized by multiple measurements, both scenarios can be addressed using mixture analysis, which expresses the data matrix \mathbf{X} as a product of two smaller matrices \mathbf{C} and \mathbf{P} . The matrix \mathbf{P} contains the chemical profiles of the source materials. In the environmental scenario these source profiles are “signatures” which might implicate one of several possible pollutants. In the chromatography scenario they might be used to identify the co-elutes via a spectral library search. The \mathbf{C} matrix contains information about the composition of the mixtures, that is, the amount of each source present in a sample.

In curve deconvolution the multiple measurement requirement implies a multichannel detector, *e.g.*, an FT-IR, UV-Vis diode array, or mass spectrometer. The samples (that is,

the rows of the \mathbf{X}) consist of the detector signal vector acquired as the peak elutes after a single injection. Each sample (row) corresponds to a different elution time. Obviously, these samples are temporally ordered. A kinetics experiment with multichannel detection is another example of this type of mixture data. In these scenarios, appropriate results are generated simply by running a mixture analysis algorithm to produce \mathbf{C} and \mathbf{P} ; no subsequent predictions are expected.

In source apportionment, the samples are usually discrete entities collected and analyzed to yield multiple measurements. These samples are unrelated by time and it may be desirable to make predictions about future samples using the profiles determined during modeling.

The two scenarios also differ in pretreatment requirements. For curve deconvolution of, say, a fused peak in a GC/MS experiment, no transforms are generally needed. The signal intensities from which sample amounts are computed correspond to those in the original data. However, in source apportionment, normalizing the \mathbf{X} data is usually necessary because of sampling effects like dilution. Area normalization is recommended, that is, transform with Divide By using the Sample 1-norm.

Note: *The nomenclature can be confusing, mostly because it employs words having multiple meanings to chemists, e.g., samples. In this chapter the terms sources, pures, and components are often interchanged. Because spectrometers are often the profile generators, the terms profile and spectra are also interchanged. The term shape is employed to describe any vector from the data matrix; it may be a row or column vector.*

A bilinear data matrix \mathbf{X} having n rows and m columns can be expressed as a product of two matrices

$$\mathbf{X} = \mathbf{C}\mathbf{P} + \mathbf{E} \quad [8.1]$$

where \mathbf{C} is a matrix of source compositions (with n rows and q columns), \mathbf{P} is a matrix of source profiles (with q rows and m columns), and \mathbf{E} is a matrix of residuals. Each element in \mathbf{X} can be found from

$$x_{ij} = \sum c_{ik}p_{kj} + e_{ij} \quad [8.2]$$

where i is the sample index, j is the variable index and k indexes the number of sources.

If a single compound is present, its measurement profile (spectrum, chromatogram, etc.) can be expressed as the vector \mathbf{p} whose values $p_1, p_2 \dots p_m$ are the intensities at each measured variable (wavelength, scan number, etc.). Each sample which is measured has a relative abundance c_i , and the compositions of a collection of samples can be expressed in a vector $c_1, c_2 \dots c_n$.

As described in “[Vectors and Matrices](#)” in [Chapter 17](#), the data matrix \mathbf{X} of [equation 8.1](#) is composed of a series of row vectors, one for each sample. For a single component source, each row has the same profile, differing only in magnitude. If two different components ($q = 2$) are present in the material analyzed, then \mathbf{C} and \mathbf{P} become matrices.

Because \mathbf{X} can be decomposed into an infinite number of \mathbf{C} and \mathbf{P} pairs, the various mixture analysis techniques differ in how \mathbf{C} and \mathbf{P} are computed. Two common mixture analysis approaches, Alternating Least Squares (ALS)¹ and Self Modeling Curve Resolution (SMCR)², are discussed below. SMCR is often called Multivariate Curve Resolution (MCR); we also adopt this terminology.

Alternating Least Squares

The premise of ALS is fairly straightforward. Given an initial estimate of the source composition (or source profiles) that contribute to the data matrix, project that estimate onto the original data to produce an estimate of the profiles (or compositions). Alternate these projections and constrain \mathbf{C} and \mathbf{P} , until a pair is found whose product $\hat{\mathbf{X}}$ is a sufficiently good approximation of the original data matrix \mathbf{X} . In the discussion that follows, estimates of quantities are shown without hats to simplify the notation.

MATHEMATICAL BACKGROUND

Number of sources

The number of source components in a mixture is related to the magnitude of variance. However, an estimation of the number of significant factors based on variance may overestimate the number of sources because some variation may come from background, baseline, noise sources, etc. It is recommended that the algorithm be repeated with the most reasonable choices for number of sources, choosing the number that yields the most meaningful solutions, that is, the best fitting compositions and profiles.

Initial estimates

Iterative techniques require initial estimates. An initial estimate of either the q purest profiles or the q purest composition vectors is required. If all but one of the sources lack a response at a given variable, that variable is a good choice for one of the initial composition vectors because it will be of a single, pure component. Similarly, if one sample is composed of only one source, that row would be a good initial profile estimate.

Unfortunately, most real data sets contain neither unique variables nor samples, thus pure shapes cannot be identified. Instead Pirouette finds the q purest shapes in \mathbf{X} , using an algorithm based on “convexity”³. These shapes may be determined row-wise, producing estimates of profiles, or column-wise, estimating composition.

Least squares optimization

If the initial estimates are of the purest variables, that is, they estimate \mathbf{C} , then \mathbf{P} is estimated from

$$\mathbf{P} = \mathbf{C}^+ \mathbf{X} \quad [8.3]$$

where the $^+$ symbol indicates the pseudo inverse of a matrix. This profile estimate is constrained to yield $\tilde{\mathbf{P}}$ with \sim indicating a constrained matrix. A new estimate of \mathbf{C} is obtained from

$$\mathbf{C} = \mathbf{X} \tilde{\mathbf{P}}^+ \quad [8.4]$$

This estimate of the amounts is constrained to yield $\tilde{\mathbf{C}}$. The product $\tilde{\mathbf{C}} \tilde{\mathbf{P}}$ is then compared to \mathbf{X} . If the approximation of \mathbf{X} is not sufficiently good, the iterative process is repeated, starting with the constrained estimate of \mathbf{C} .

ALS has a certain symmetry. If the initial estimates are of the purest samples (*i.e.*, estimating \mathbf{P}), the regression in [equation 8.4](#) occurs first, followed by the regression in [equation 8.3](#).

Constraints

Currently, Pirouette supports several types of constraints.

Non-negativity. The fractional amount of a source in a mixture must be 0 or greater. This allows an assumption of non-negativity in the composition matrix. Similarly, the intensities produced by many measurement techniques cannot be negative, leading to non-negativity constraints on values in the profiles. Exceptions to the latter include such measurements as circular dichroism spectra and spectra which have been derivatized; in these cases, non-negativity should not be applied.

Unimodality. A single compound in chromatographic analysis elutes as a single peak; the peak is “unimodal”. A bimodal peak, that is, a peak with two maxima, would be considered to be composed of more than one compound. Thus, it is reasonable to constrain the compositions to be unimodal for curve deconvolution data.

Closure. For some experiments, all amounts are expected to sum to a constant; the amounts are said to be closed. Closure may be applied to either compositions or profiles. If neither is closed, the final profiles are automatically normalized as ALS processing completes.

RUNNING ALS

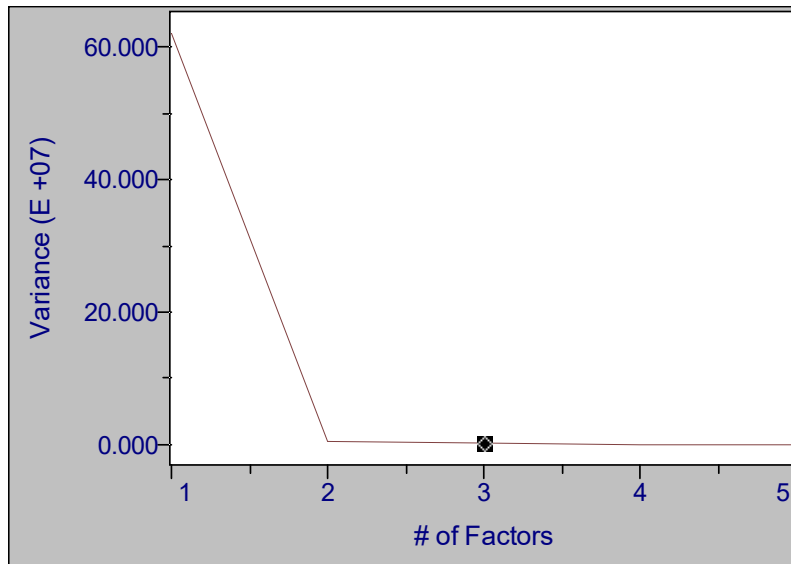
ALS offers a number of choices for constraints applied during the least squares optimization (see “ALS options” on page 16-26) as well as the maximum number of components to consider. Keep in mind that constraining profiles to be non-negative conflicts with derivatives or with transforms that by their nature create some negative values. This is also true if mean-centering or autoscale are selected as pre-processing options because the result will have both positive and negative values.

Objects computed during ALS include the PCA Scores, Loadings and Eigenvalues (see “Running PCA” on page 5-31), plus several objects which aid in interpretation of the algorithm results.

Source Select

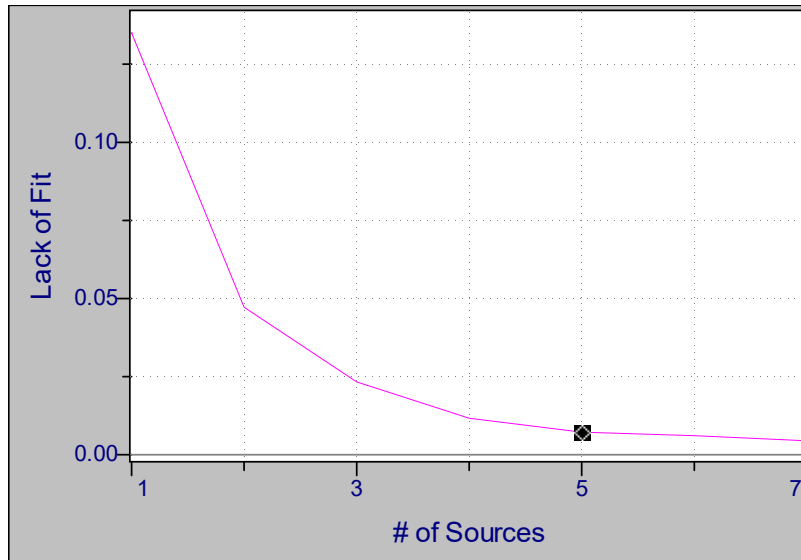
The eigenvalues are shown in a plot called Source Select. A diamond “handle” on this plot indicates the initial number of sources estimated by Pirouette. Manipulating the position of the diamond triggers recalculation of amounts and profiles.

Figure 8.1
ALS Solution Select
object



The variance shown in this plot will often be concentrated in the first factor when no pre-processing is performed before the algorithm. This can make deciding how many sources are present a challenge. Another metric which may help is a Lack of Fit computation based on the X Residuals.

Figure 8.2
ALS Lack of Fit



Source Amounts and Source Profiles

The final estimates of the pure source compositions and profiles depend on the Source Select setting.

Figure 8.3
ALS Source amounts
object

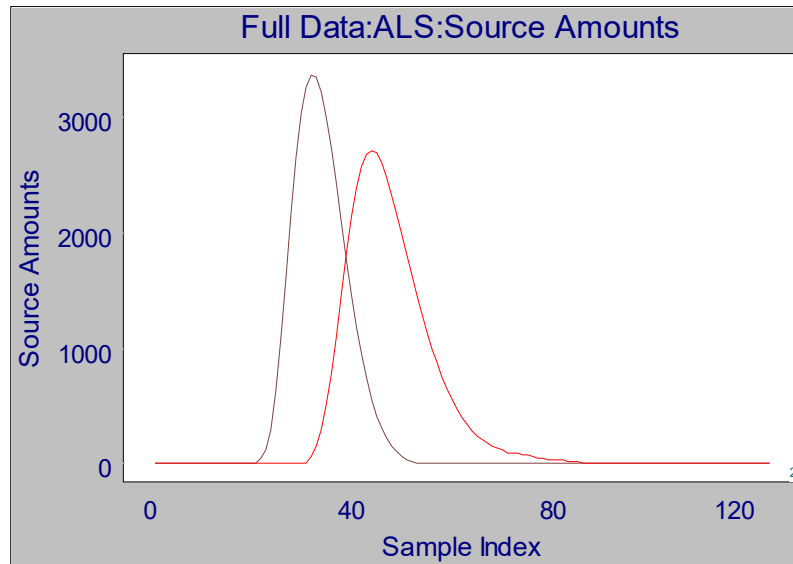
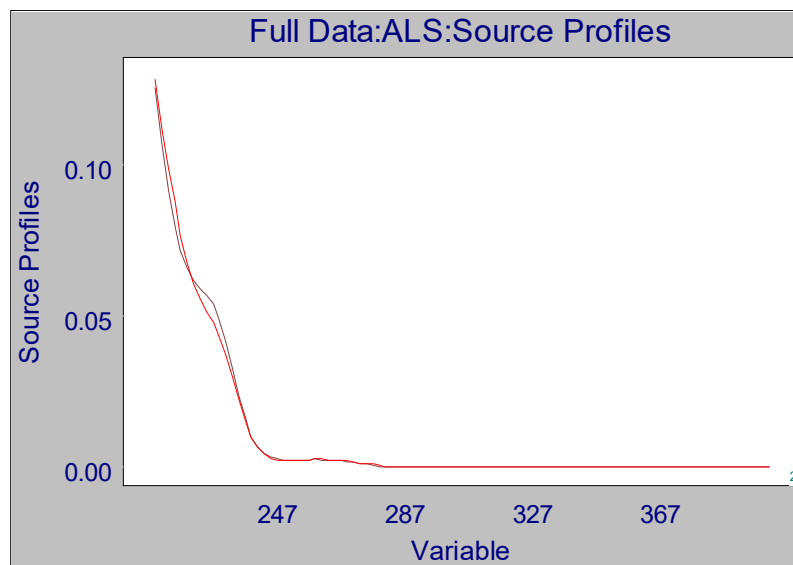


Figure 8.4
ALS Source profiles
object



To contrast the final solutions with the initial estimates, examine the corresponding initial amounts and initial profiles objects.

Figure 8.5
Initial Amounts
object

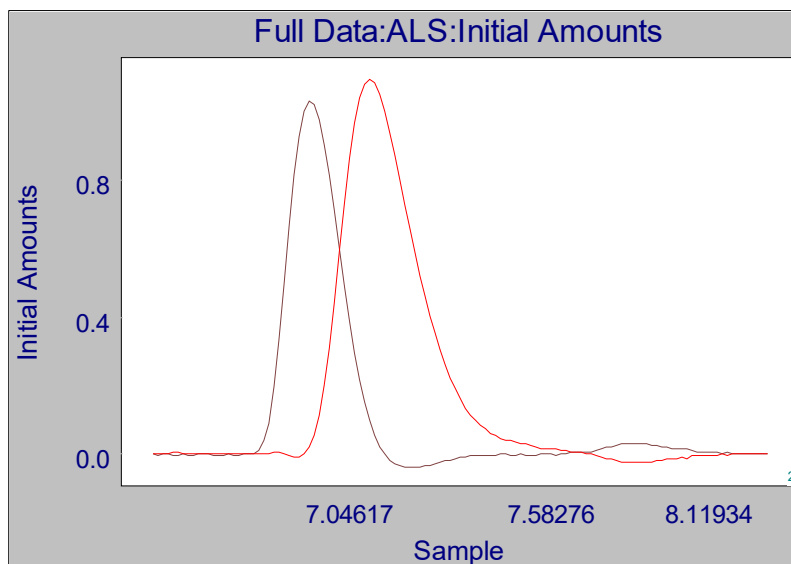
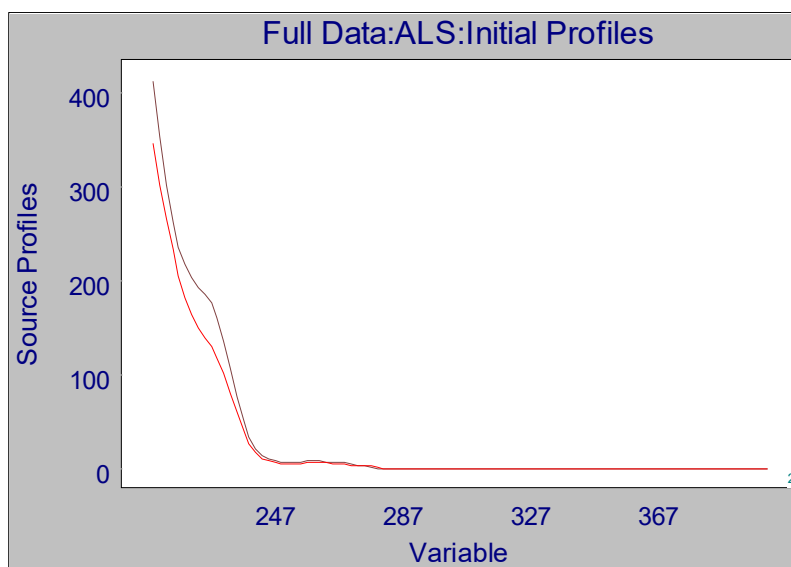
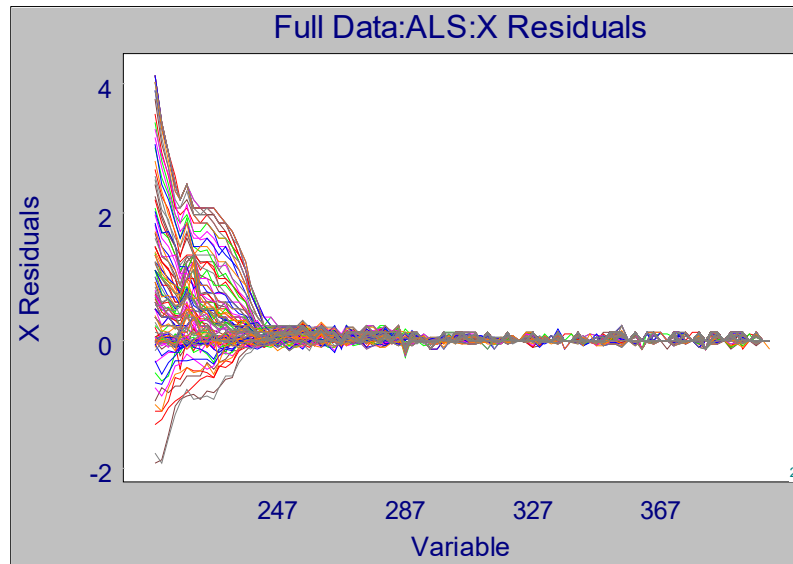


Figure 8.6
Initial Profiles
object



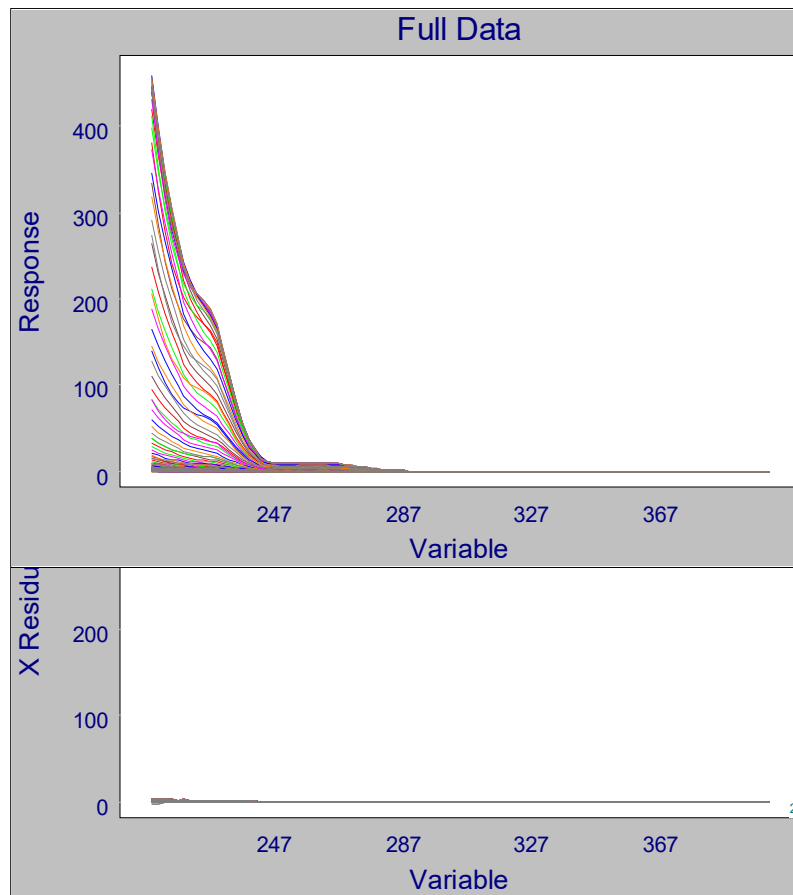
The X Residuals reveals which portions of the original data remain unmodeled.

Figure 8.7
ALS X Residuals
object



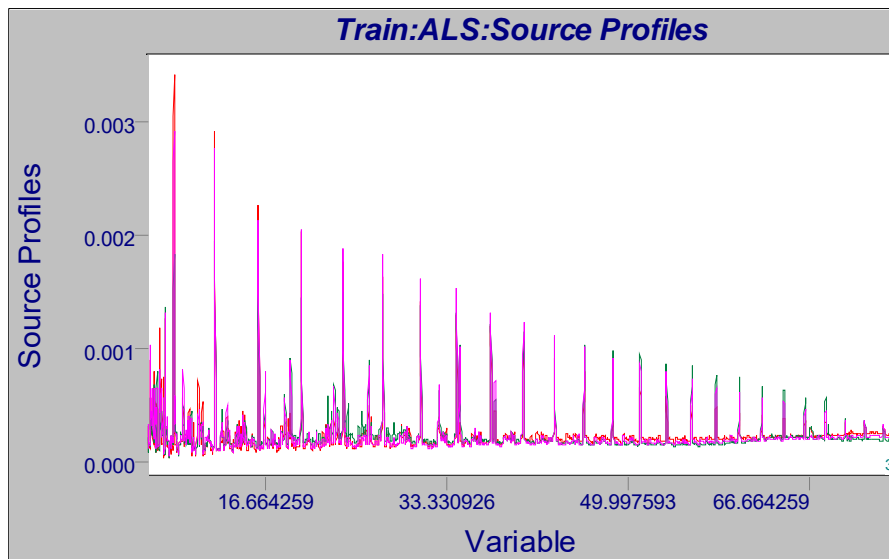
The magnitude of the X Residuals relative to the original (or transformed) X data can be inferred by plotting the (transformed) X data and the X Residuals on the same scale. Choose Display > Limits to present a dialog (see “Magnifying Regions” on page 12-15) setting limits for the X and Y axes.

Figure 8.8
X Residuals scaled
to the same
magnitude as X



For source apportionment applications, it is customary to first normalize the samples to a common scale: Divide By, Sample 1-norm. In the following example, the data are a set of chromatographic profiles.

Figure 8.9
ALS source apportioned profiles



Note: In the case of chromatographic data, it may also be necessary to align the chromatographic profiles before normalizing (“Align” on page 4-22).

The mixture compositions in a source apportionment application may be best evaluated in tabular format, as shown below.

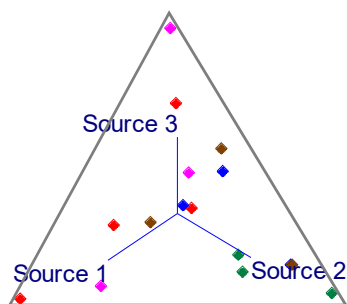
Note: It may be useful to apply an additional transform after normalization—Multiply, by 100—to put the source compositions on a percent rather than fractional scale.

Figure 8.10
ALS source apportioned amounts

		1	2	3
		Source 1	Source 2	Source 3
1	SIG20802	99.6825	0.0000	0.9019
2	SIG20803	0.3420	99.5443	0.0000
3	SIG20804	0.3698	0.1304	99.4493
4	SIG20805	30.4098	35.5028	33.7991
5	SIG20806	43.9087	28.0080	27.7035
6	SIG20811	56.0209	16.5489	27.2445
7	SIG20812	24.2440	66.9633	9.1286
8	SIG20813	71.5869	23.3797	4.7397
9	SIG20814	7.9782	81.2306	10.6722
10	SIG20815	7.6675	81.4279	10.9321
11	SIG20816	13.3865	15.1951	71.3063
12	SIG20817	22.8361	62.4196	14.8819

A scatter plot can illustrate the relationships among the source contributions. The following 3D scatter plot shows the results from the 3-source solution of [Figure 8.10](#) superimposed over a triangle much like that of a phase diagram. The Source axes point in directions of increasing amounts of each source, and points in the interior of the triangular region represent mixtures of all 3 sources.

Figure 8.11
ALS source amount
ternary diagram



Note that a 3-source solution can be depicted in a 2 dimensions like that shown above. A 4-source solution could be illustrated in the 3-D plot available in Pirouette. More than 4 sources cannot easily be shown graphically in a single plot. Instead, show these results in a multiplot.

MAKING AN ALS PREDICTION

To apply the source profiles computed from one data set to new data in order to predict amounts in those new samples,

- Load a new data set
- Load the ALS model
- Choose Process > Predict

In the Predict Configure dialog, click on the ALS model name to show information about the model in the adjacent info box.

Figure 8.12
ALS Model info

```
Model Info:
ALS (v. 5.00)
Created: 01/02/2023 15:57:45.257
User ID: n/a
ID Source: n/a
Windows Login: BGRDELL:Brian

Source file: ternary.dat
Exclusion set: Full Data

Sample count: 16
X variable count
Total: 4203
Included: 4203
First X name: 4.730926
Last X name: 74.764259

Transforms:
  Aligned to 1 (window = 51)

Preprocessing: None
Validation: None
Assumed # of Sources: 3
Closure Choice: Amounts
Non-negativity constraints: Amounts & Profiles
Unimodality constraints: None
```

After verifying that the model is appropriate, choose the subset on which to perform the prediction and click the Run button. When processing is completed, three computed objects are available for inspection.

Source Profiles

The Source Profiles used in an ALS prediction are those stored in the ALS model and are shown for completeness.

Source Amounts

An estimate of the predicted Amounts is computed by multiplying the Prediction target X data by the pseudo inverse of the modeling profiles P . The C constraints specified when the model was built are then applied to produce the values shown in the Source Amounts object.

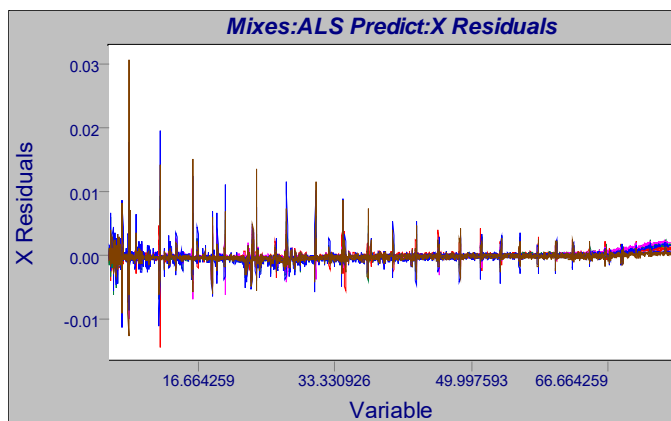
Figure 8.13
ALS predicted
source amounts

17,4

		1	1	3
		Source 1	Source 2	Source 3
1	SIG20814	80.8985	8.4303	10.6654
2	SIG20815	81.2758	8.4360	10.4532
3	SIG20816	15.7407	14.4390	69.9492
4	SIG20817	62.7376	24.7126	12.8405
5	SIG20818	32.0019	22.8334	44.9945
6	SIG20819	43.6539	14.3876	42.7605
7	SIG20820	39.2062	10.2761	51.5190
8	SIG20821	39.1306	28.2899	32.3368

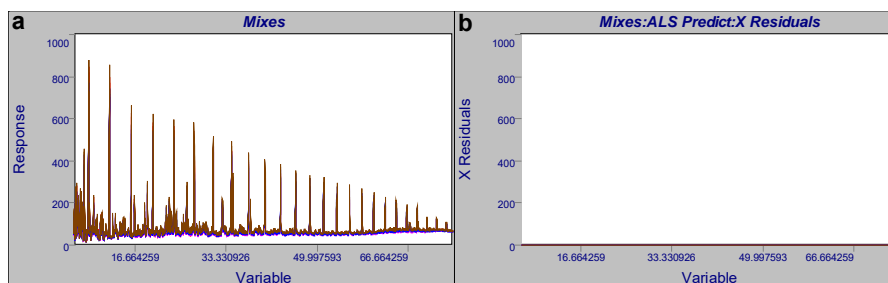
The prediction X Residuals illustrate the errors in reconstructing the prediction data via equation 8.1. These residuals as a line plot reveal structure in regions less well modeled.

Figure 8.14
ALS prediction X
Residuals



It may be worthwhile scaling the X Residuals plot to the scale of the original data to observe the magnitude of the residuals in contrast with the magnitude of the raw data (see following figures).

Figure 8.15
Comparison of a) X
and b) prediction X
Residuals on the
same scale; the latter
are not detectable



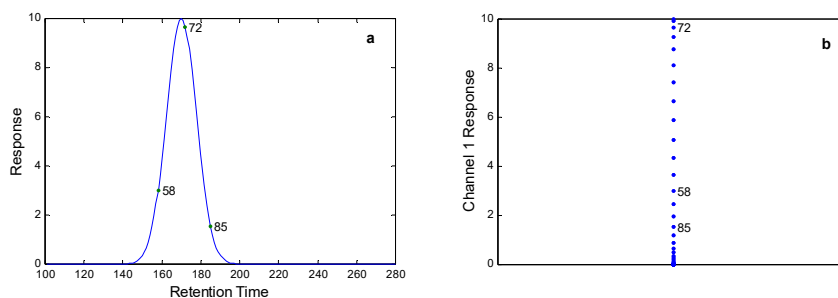
Multivariate Curve Resolution

Understanding MCR is facilitated by a series of chromatographic examples, starting with single channel detection for a single component peak and progressing to two, three, and

finally many channel detection for two component peaks. Note that the current implementation of MCR in Pirouette is limited to computations of two sources.

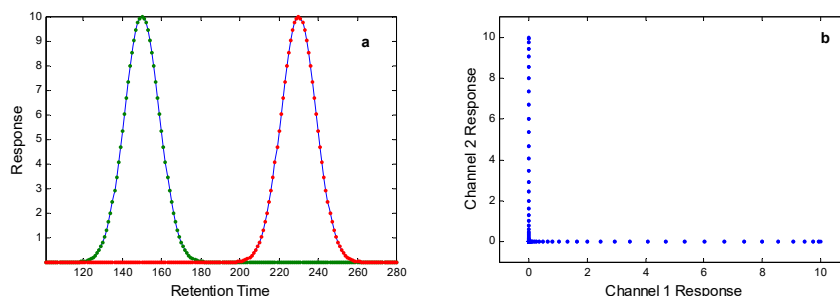
As a pure compound elutes from a column, the response of a single channel detector is a function of the compound's concentration in the mobile phase. Figure 8.16a shows the detector response as a function of time. Note, however, that the detector response itself, Figure 8.16b, is one-dimensional, that is, it rises and falls with concentration, but no other discriminating information is present other than the magnitude of the maximum response. If the peak contains two overlapped compounds, the results are identical: univariate detection cannot discriminate between the two underlying components.

Figure 8.16
Single channel
detection
of a pure peak;
(a) chromatogram
(b) biplot of detector
responses; three
points are labeled



Now consider two channel detection with each compound responding to one channel only. If the two compounds are completely resolved, a chromatogram like that in Figure 8.17a is produced; Figure 8.17b shows the two detector responses plotted against each other. In this idealized scenario, having more than one detector channel reveals that at least two components have eluted and that their responses are practically identical.

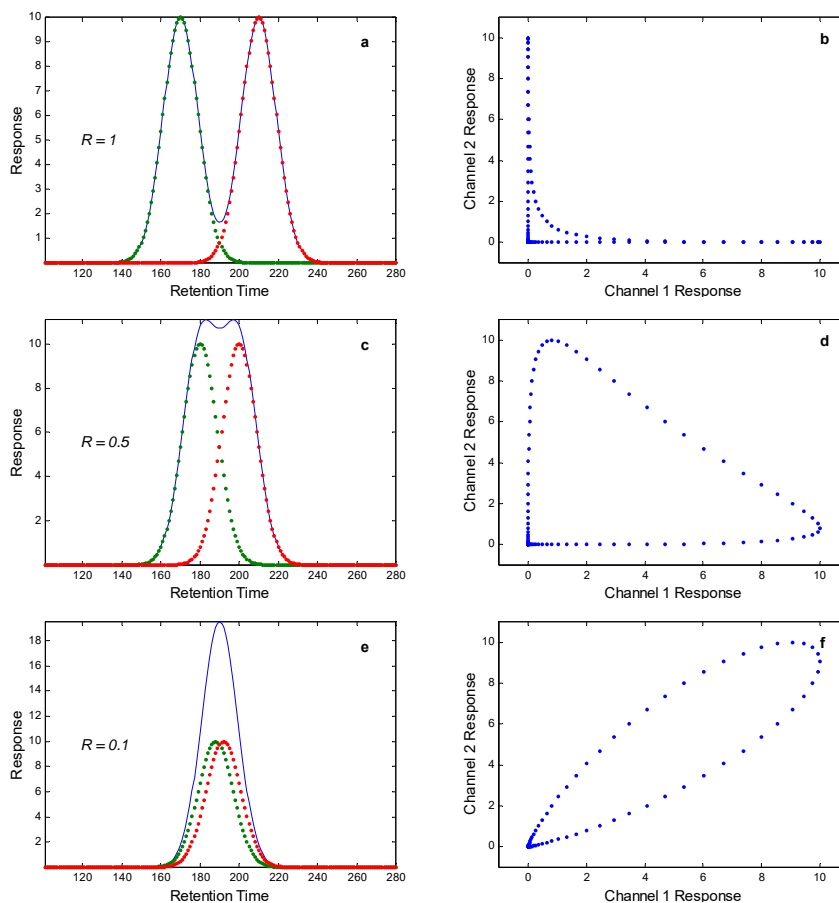
Figure 8.17
Two component
mixture, two channel
detection, no
overlap:
(a) chromatogram
(b) biplot of detector
responses



It is instructive to compare biplots as chromatographic resolution decreases; several conditions are illustrated in Figure 8.18 below.

Even for incomplete separation, the two-channel system still reveals the number of components in the fused peak. Moreover, the failure of the points in the middle of the peak (*i.e.*, between the component peak tops) to return to the origin in all biplots implies component overlap. Points deviate from the perpendicular axes when their spectral response no longer arises from one component only. For example, in Figure 8.18b, the points in the biplot begin to deviate from the vertical “pure” axis after the first component’s peak maximum is reached—the greatest excursion from the origin—but long before the response level for the first component has returned to zero. This signals the start of elution of the second component as denoted by the chromatographic trace on the left.

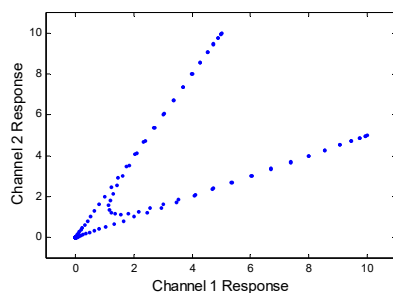
Figure 8.18
Two component mixtures at different resolutions R : (a), (c) and (e) chromatograms; (b), (d) and (f) biplots of detector responses



A similar interpretation of the second and third examples in Figure 8.18 indicates that the second component appears soon after the first component begins to elute.

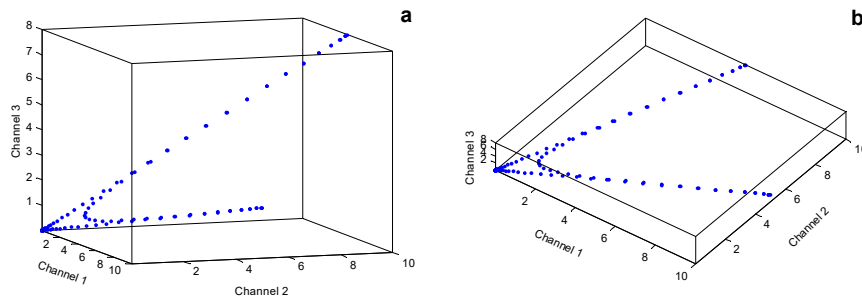
Of course, selective detection channels are rare in the real world; an analyte typically generates a response at many or all channels. In that case the “pure” points no longer coincide with the axes for each channel in the biplot. Instead the angle of the vector for each pure component is determined by its relative response at the two channels. In a mixture peak of nearly complete resolution, such as that in Figure 8.19, the biplot of non-selective channel responses still shows the nature of the overlap and of the detector response differences of the two components.

Figure 8.19
Two component mixture, two channel non-selective detection; $R = 1.0$



If the detector has more than 2 channels, interpretation is much the same as before, although we must think about more than 2 dimensions. The following figure shows a plot of the data from Figure 8.19 with a 3-channel detector.

Figure 8.20
Two component
mixture, three
channel non-
selective detection;
 $R = 1.0$

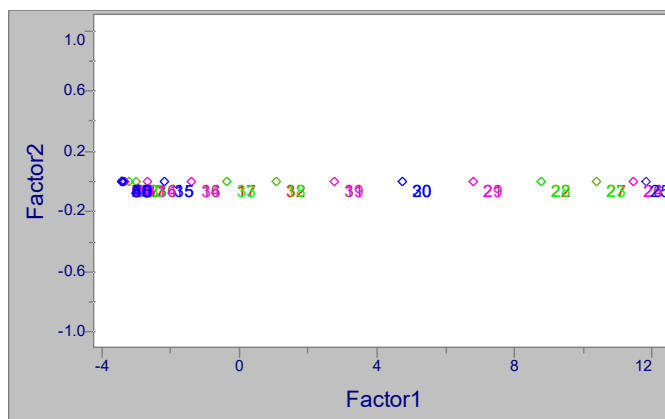


Because only two analytes are present, all points actually lie in a plane. This is also true even when the number of detector channels is very large. Rotating the data points into a different perspective (see Figure 8.20b) produces a two-dimensional image essentially identical to those described in Figure 8.19. This means of simplification, often used to facilitate interpretation of multivariate data, is accomplished by Principal Components Analysis. Thus, MCR starts with a PCA decomposition.

MATHEMATICAL BACKGROUND

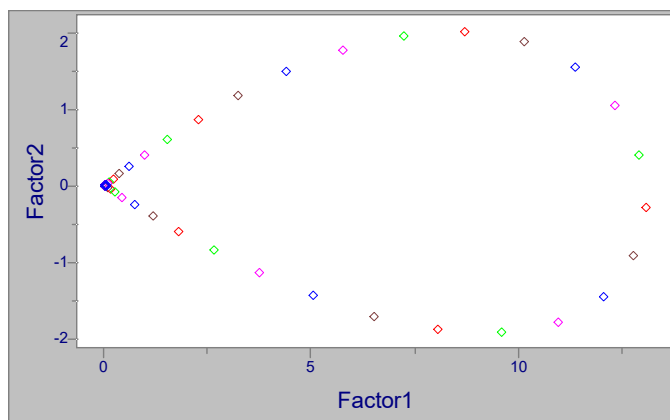
Consider again a single-component peak. The spectra collected at several time points across the peak differ only in intensity and noise contribution. PCA of the time-spectrum data matrix produces only one significant eigenvalue and a scores plot with all points along a ray.

Figure 8.21
Scores of a single
component peak;
labels denote sample
number



Now suppose that two components coelute. At time points across the peak, the data matrix has contributions from both components. Assuming linear additivity in the detector, the observed spectrum at any time is the sum of proportional amounts of the individual source spectra, where the proportions are the relative amounts of each source at that elution time. If the noise in this mixture region is negligible, PCA produces only two significant eigenvalues; the data points from the overlap region fall in a space defined by two factors. The scores plot from such an overlapped chromatogram is shown below. Note its similarity to Figure 8.18f; both exhibit a shape characteristic of a two source mixture.

Figure 8.22
Scores from a two
component peak



The points start off along a ray defining the spectrum of one component; they then bend around to another ray, the spectrum of the other component. The curved portion of the plot contains the mixture points.

A matrix \mathbf{X} can be decomposed into PCA scores and loadings (see “Mathematical Background” on page 5-16):

$$\mathbf{X} = \mathbf{TL}^T \quad [8.5]$$

For a two factor matrix derived from a two component mixture, each spectrum x_i can be expressed as:

$$x_i = t_{1i}L_1^T + t_{2i}L_2^T \quad [8.6]$$

Since all of the information about the overlapped peak is in this factor space, then the spectra of the pure components can also be expressed in this manner:

$$P_1 = \rho_{11}L_1^T + \rho_{12}L_2^T \quad [8.7]$$

$$P_2 = \rho_{21}L_1^T + \rho_{22}L_2^T \quad [8.8]$$

where P_1 and P_2 are the spectra of the two pure components, and (ρ_{11}, ρ_{12}) and (ρ_{21}, ρ_{22}) are their corresponding scores. In other words, the spectra of the mixture samples, as well as the pure compounds, can be expressed as a function of their scores and loadings defining that factor space.

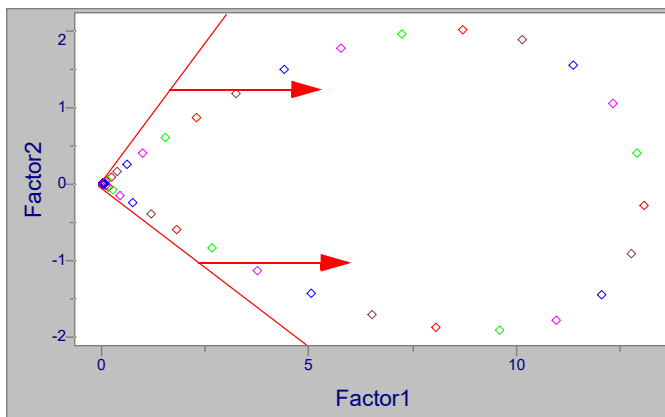
If by observation of the scores in the factor space we can identify (ρ_{11}, ρ_{12}) and (ρ_{21}, ρ_{22}) , then the underlying pure component spectra can be determined from [equation 8.7](#) and [equation 8.8](#). In general, it cannot be assumed that any samples (rows of \mathbf{X}) are instances of a pure component. Determining possible pure spectra is the goal of MCR.

Certain mathematical relationships must hold for all proposed pure spectra. First, all the intensities in a proposed spectrum must be either positive or zero. This implies that any candidate score (τ_1, τ_2) must satisfy:

$$\tau_1 l_{1j} + \tau_2 l_{2j} \geq 0 \quad \text{for all } j \quad [8.9]$$

where l_{ij} are elements of loading L_i . This restriction is illustrated by a pie-shaped region in the scores plot shown below. Only points inside the wedge have positive or zero intensities in the elements of the spectral vector.

Figure 8.23
Non-negative
response restriction

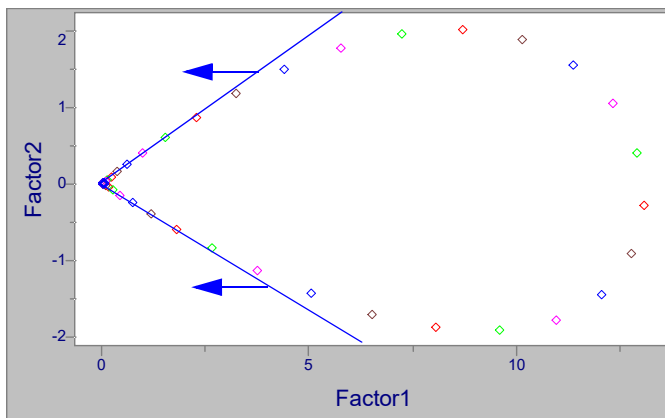


Second, each mixture spectrum is a linear combination of the pure components:

$$x_i = \alpha_i P_1 + \beta_i P_2 \quad [8.10]$$

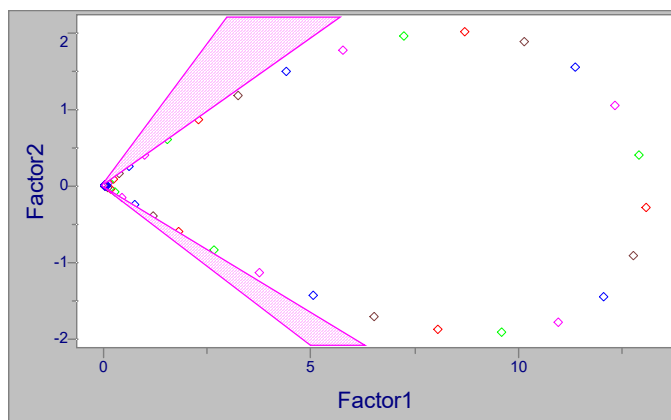
Since the amounts of the pure components must be either positive or zero, then $\alpha_i \geq 0$ and $\beta_i \geq 0$. This translates to an exclusion of the region in factor space bounded by the most extreme samples in the mixture as shown in the following figure.

Figure 8.24
Non-negative
composition
restriction



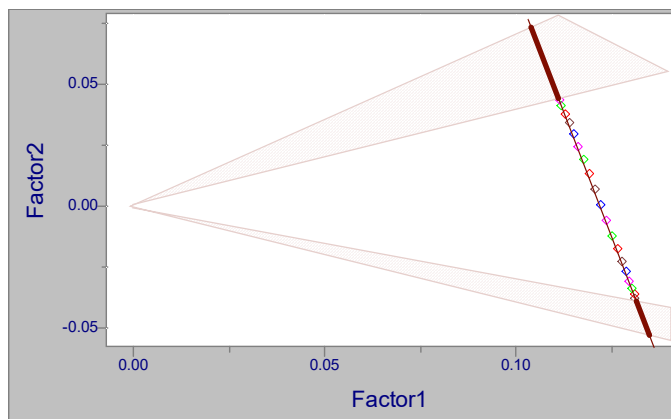
Points inside this wedge have negative pure components amounts; only points outside the edge satisfy this second criterion. Thus, the pure component spectra must lie in the shaded region shown below, which is the intersection of the allowed regions in the two previous plots.

Figure 8.25
Non-negativity
constraints
combined



A final restriction can be imposed by normalizing the spectra to unit area. This places the mixture points on a straight line, as shown below. Note that the scale is not the same as in the previous plot.

Figure 8.26
Two-component
curve resolution
constraints



The thick portions of the line represent mathematically feasible solutions for the two pure component spectra, one falling on the thick upper line, the other falling on the lower thick line. If the spectrum of one component has a unique channel, that is, a response totally absent from the other spectrum, then a unique solution exists; the thick line collapses to a point. Otherwise, the user is left to choose the “best” solution in the feasible region.

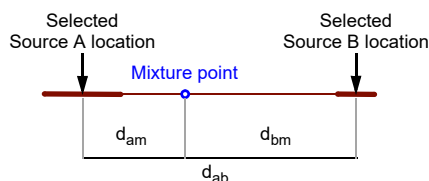
How to choose is a topic of debate in the MCR community. Some propose the feasible region midpoint as the most reasonable solution when nothing is known about the underlying components. Others suggest the points at the extremes of each of the lines (known as the “outer bounds”) since these are the most dissimilar of the mathematically feasible spectra. Still others prefer the “inner bounds” because these are the most pure spectra in the data set.

If the user knows something about the spectral shapes, then observation of the possible resolved spectra can lead to an appropriate selection of the best location along the feasible lines. Similarly, the chromatographer might apply knowledge about peak shapes to further constrain the solution.

Once a location is selected in the feasible region, the profile shapes can be computed from the coordinates of the location. These coordinates are the scores for the points; multiplying by the eigenvectors will yield the normalized source profiles as in [equation 8.7](#).

The relative source composition at any location can be inferred as from a mole fraction diagram. For example, in the figure below, the solution line of Figure 8.26 has been rotated for simplicity. The source locations have been fixed, and the location of one of the samples in X is shown as a mixture point. The distances of the mixture point (m) to the source locations are marked as d_{km} , where k is the source index.

Figure 8.27
Computing the
source compositions



The fractional compositions are computed from:

$$C_{ma} = \frac{d_{bm}}{d_{ab}} \quad C_{mb} = \frac{d_{am}}{d_{ab}} \quad [8.11]$$

The fractional compositions are scaled to the final amounts by comparing to the total intensities found in the X data, found by summing each column.

RUNNING MCR

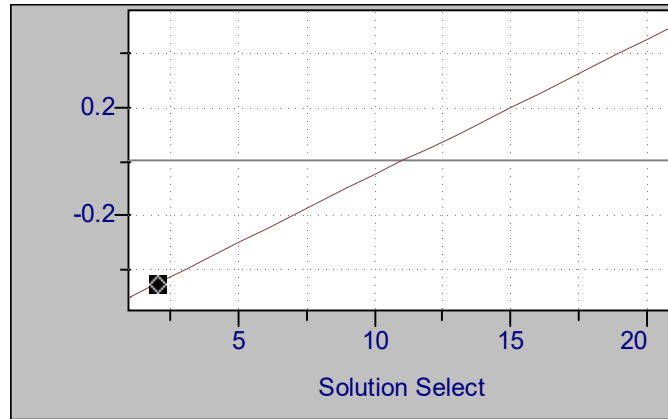
The two component curve resolution algorithm currently implemented has no preprocessing options other than a choice of transforms; see “Transforms” on page 4-10. For curve deconvolution of, for example, a fused peak in a GC/MS experiment, no transforms are generally needed (note that a separate normalization is performed during the algorithm; see Figure 8.26). The signal intensities from which the MCR algorithm computes per sample amounts are those in the raw data. However, if the mixture samples are obtained independently, as in source apportionment, it is usually necessary to normalize the data because each sample derives from a different analysis. In this situation, area normalization is recommended, that is, Divide By with a Sample 1-norm option.

Objects computed during MCR include the PCA Scores, Loadings and Eigenvalues (see “Running PCA” on page 5-31), plus several objects which aid in interpretation of the algorithm results.

Solution Select

As mentioned earlier, a range of feasible solutions typically exists; a choice of location within the feasible regions produces estimates of the underlying profiles and amounts. Pirouette chooses the inner bounds of the feasible region by default but you can modify the result by manipulating the Solution Select object shown below.

Figure 8.28
Solution Select
object



This object allows you to ‘tune’ the results by clicking and dragging the diamond-shaped handle in the plot. Handle position corresponds to the relative location within the upper and lower portions of the feasible regions (see [Figure 8.26](#)). The resolved source profiles and corresponding source amounts discussed below are linked to the handle position.

Note: The solution select axis is divided into 20 segments. Position 1 represents the inner bound while position 21 is the outer bound. Plots linked to this object show the position of the solution select handle in the lower right.

Source Profiles and Source Amounts

Based on the position of the Solution Select setting, the Source Profiles and Source Amounts objects can be computed. Examples of these plots follow.

Figure 8.29
Source Profiles
object

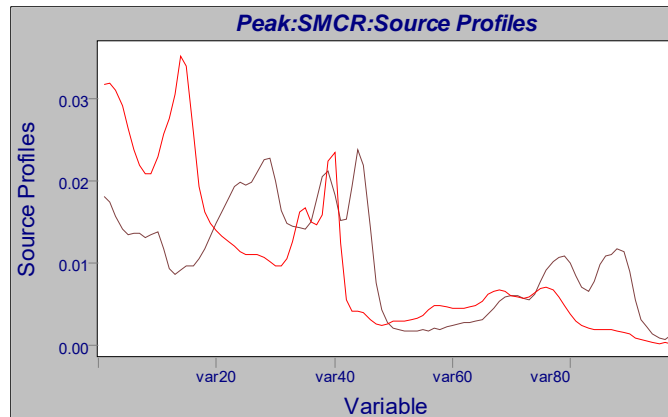
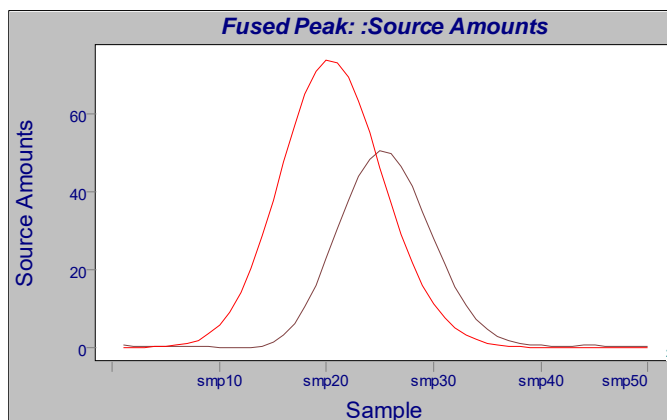


Figure 8.30
Source Amounts
object

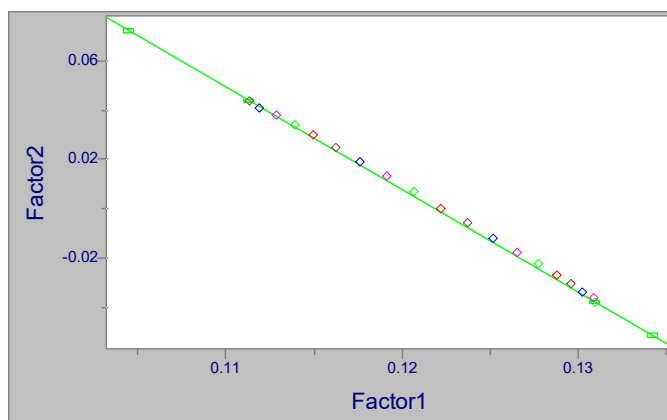


Note: The default view of Source Amounts is a Table view, but for chromatographic peak resolution applications, it is more practical to view this object as a line plot, particularly when varying the Solution Select setting.

Feasible Region

The feasible region discussed earlier (see Figure 8.26) is illustrated in Pirouette by a computed object of the same name. The inner and outer bounds are represented as a pair of small rectangles for each component.

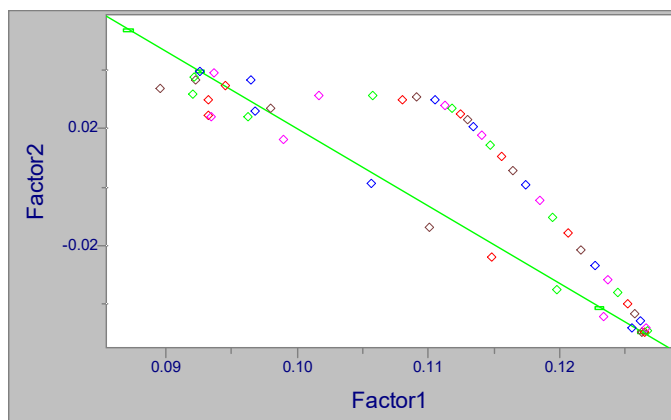
Figure 8.31
Feasible Region
object



This MCR implementation assumes only two underlying components. The user must investigate the quality of this assumption because if more than two components are present, the computed results may be unreliable. A sum of the first two PCA eigenvalues close to 100% of the variance supports the two component assumption.

The Feasible Region can also provide evidence about the veracity of the two component assumption. If a third component is present, samples may deviate from the diagonal line representing the area normalization constraint. This can occur, for example, if a noisy baseline region surrounding a fused chromatographic peak is included in the data matrix, as is shown below.

Figure 8.32
Feasible region
when more than 2
components are
present



Bounded Source Profiles and Bounded Source Amounts

The Bounded Source Profiles and Bounded Source Amounts objects overlay the profiles and amounts, respectively, for the inner and outer bounds for each component. Small differences in the extreme traces indicate weak dependence on the Solution Select setting.

Figure 8.33
Bounded Source
Profiles, for
component 1

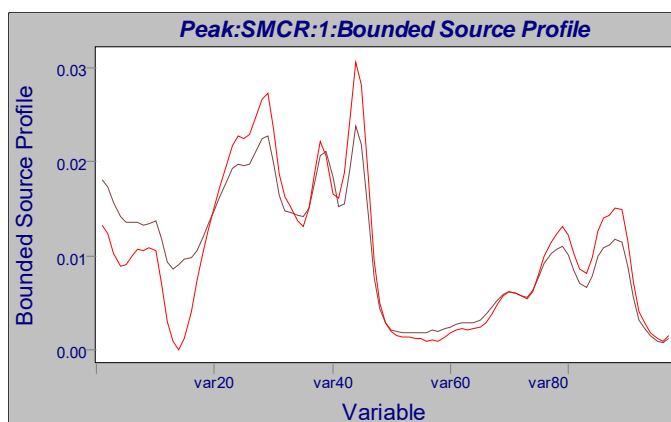
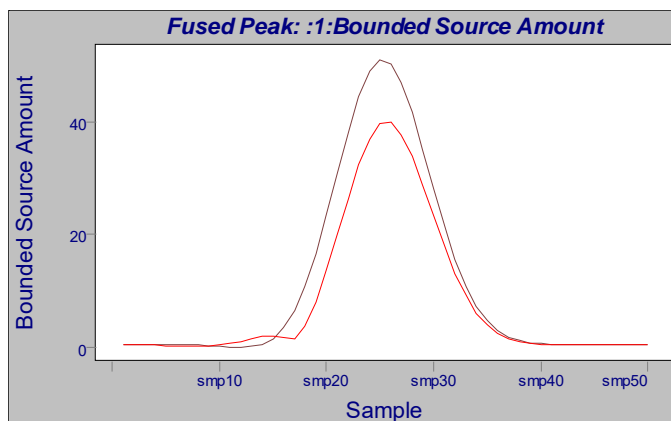


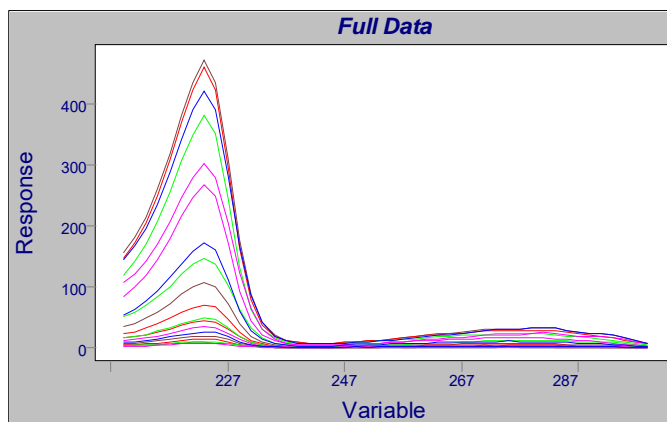
Figure 8.34
Bounded Source
Amounts, for
component 1



The example file included on the Pirouette CD—MNAPS . DAT—provides an illustration of tuning the MCR result using *a priori* knowledge, in this case, the chromatographic peak shape. Shown below are the spectra acquired across the peak; each trace corre-

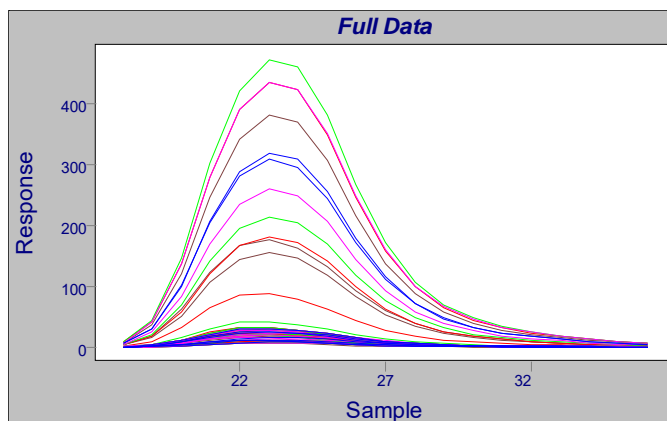
sponds to a different retention time, thus the variation in intensity. Spectral shape differences are difficult to discern.

Figure 8.35
Spectra of MNAPS
data: line plot of the
rows



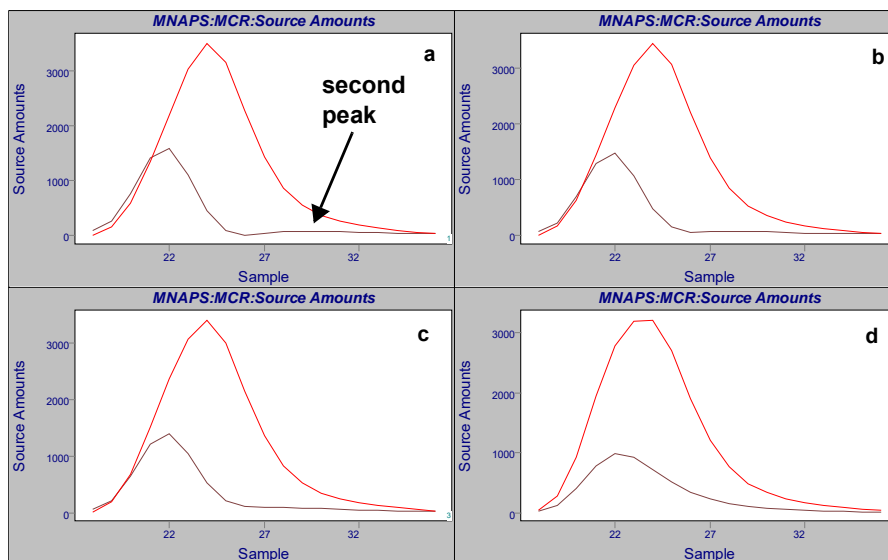
This is also true of the chromatographic peak profiles, shown next, one trace for each wavelength. Here, the heights of the peak are a function of the response in the spectra at a given wavelength. It appears that only a single component forms this peak.

Figure 8.36
Chromatographic
peaks in MNAPS
data: line plot of the
columns



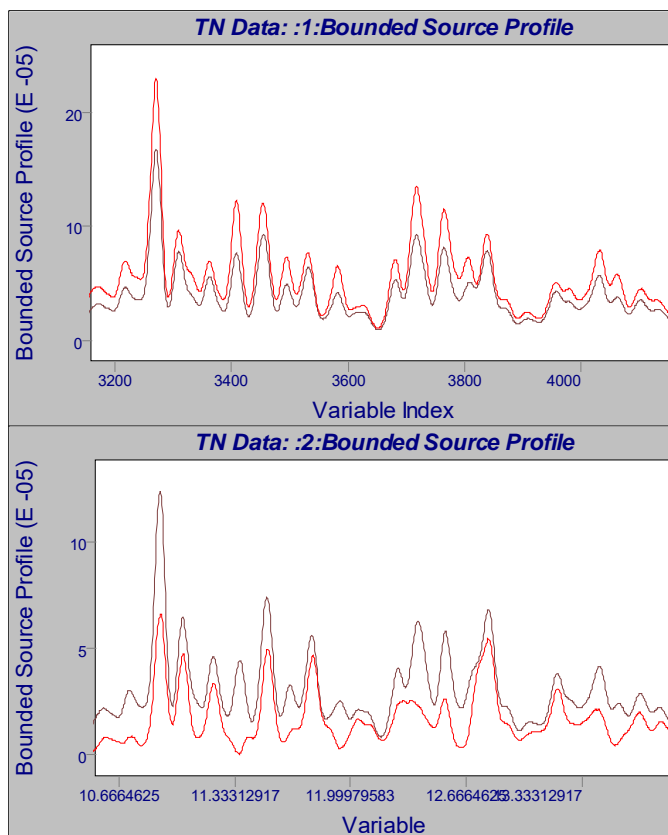
MCR applied to this fused peak produces the plots that follow. Shown in [Figure 8.37a](#) is the inner bound solution, where the profile of one peak is resolved into two peaks, an unexpected chromatographic outcome. By moving the handle in the solution select object across the solution region, different results are produced, as shown in [Figure 8.37b-d](#). The first solution that results in a single chromatographic peak is at position 3, [Figure 8.37c](#). Moving too far away from the inner bound (to higher values of the solution select position) makes the resolved peak too skewed and tailing (see [Figure 8.37d](#)). It appears that position 3 is the preferred solution.

Figure 8.37
MCR solutions for
MNAPS data;
resolved amount
profiles



The preceding example plots focus primarily on the results for curve deconvolution of a chromatographic peak. When using MCR for source apportionment, it is recommended that the samples are normalized: set the Divide By transform, using the Sample 1-norm option. This will result in resolved amounts that sum to 1. An example of such an approach is shown in the following figures.

Figure 8.38
Bounded Source
Profiles for two
mixture
components; data
were normalized
before MCR



The bounding profiles—shown in the figures as a zoomed region of the resolved chromatograms—are quite similar. The position of the solution selection handle has little effect on the outcome.

Figure 8.39
Bounded Source
Amounts for two
mixture
components; data
were normalized
before MCR

		a		b			
		1	2				
		Inner Boun	Outer Bour				
		1	2				
1	20424	1.0000	0.6244	1	20424	0.0000	0.3756
2	20425	0.0000	0.3581	2	20425	1.0000	0.6419
3	20426	0.7983	0.5713	3	20426	0.2017	0.4287
4	20427	0.6053	0.5194	4	20427	0.3947	0.4806
5	20428	0.4594	0.4803	5	20428	0.5406	0.5197
6	20429	0.4794	0.4857	6	20429	0.5206	0.5143
7	20430	0.1616	0.3919	7	20430	0.8384	0.6081

The source amounts, on the other hand, are very sensitive on the solution position. For source apportionment applications, it is customary to view the source amounts in the tabular form shown above. The inner bound solutions imply that the most extreme samples, 20424 and 20425, would be the pure components.

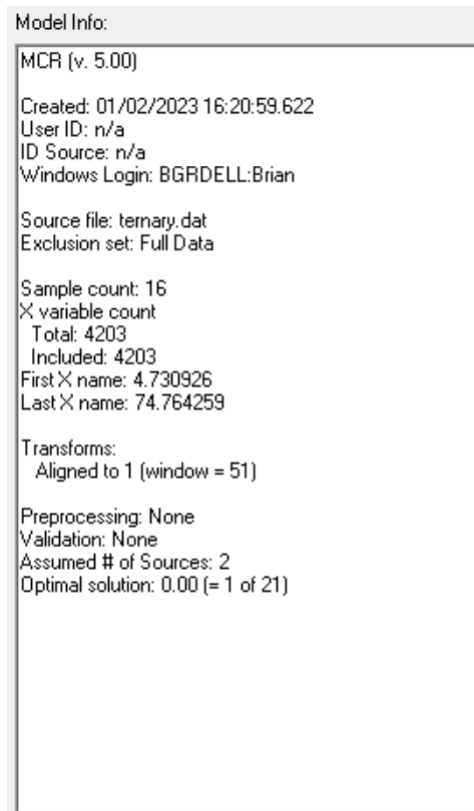
MAKING A MCR PREDICTION

To estimate the contributions to new mixture data based on the profiles derived from a prior MCR analysis,

- Load a new data set
- Load the MCR model
- Choose Process > Predict

In the Predict Configure dialog box, click on the MCR model name to show information about it.

Figure 8.40
MCR Model info

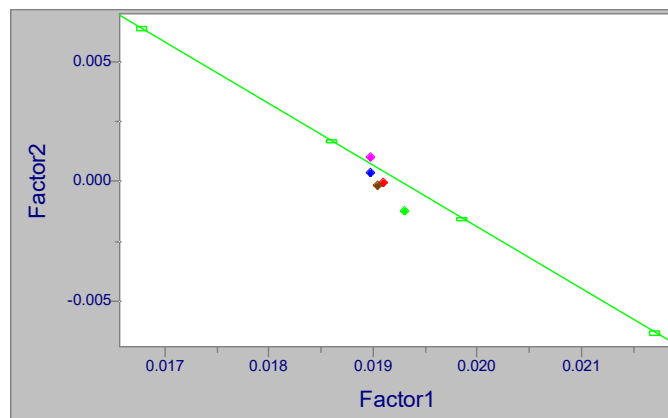


After verifying that this is the correct model for processing the new mixture data, click on the data set name, then on Run to start the prediction processing. Two computed objects result.

Feasible Region

The feasible region for prediction data is a projection of the new data set onto the factor space of the training set, shown with the upper and lower bounds from the training data.

Figure 8.41
MCR prediction
feasible region



Source Amounts

The amounts of each training set source profile in each prediction sample are shown, by default, as a tabular view, such as the in the following figure.

Figure 8.42
MCR prediction
source amounts

27,4		1	1
		Source 1	Source 2
1	20426	0.7874	0.2126
2	20427	0.6124	0.3876
3	20428	0.4711	0.5289
4	20429	0.4918	0.5082
5	20430	0.1784	0.8216

Reference

1. Tauler, R.; Kowalski, B.R.; and Fleming, S. "Multivariate curve resolution applied to spectral data from multiple runs of an industrial process". *Analytical Chemistry*. 1993; 65(15): 2040-2047.
2. Lawton, W. H. and Sylvestre, E. A. "Self modeling curve resolution". *Technometrics*. 1971; 13(3):617-633.
3. Grande, B.-V. and Manne, R. "Use of convexity for finding pure variables in two-way data from mixtures". *Chemometrics and Intelligent Laboratory Systems*. 2000; 50:19-33.

Examples

Contents

Description of Example Files	9-1
Food and Beverage Applications	9-5
Environmental Science Applications	9-11
Chemometrics in Chromatography	9-17

This chapter describes the files included with Pirouette and three different multivariate application areas. The example files are a means to familiarize you with the operation of the software and demonstrate general chemometric principles. Several figure prominently in the tutorials in [Chapter 2, Pattern Recognition Tutorial](#) and [Chapter 3, Regression Tutorial](#). Following the file description are discussions of how the multivariate technology has been implemented in various fields. The overview topics include

- “Food and Beverage Applications”
- “Environmental Science Applications” and
- “Chemometrics in Chromatography”

References are supplied at the end of each overview.

Description of Example Files

[Table 9.1](#) summarizes both the data source and possible direction of multivariate investigations for the files included on the Pirouette distribution disk. Pertinent details of each file are then provided.

Table 9.1
Example Files

Name	Field	Data Source	Multivariate Issue
ALCOHOL	Clinical	Wet chemistry	Classifying disease state
ARCH	Archaeology	X-ray fluorescence	Classifying stone artifacts
BUTTER	Food	Headspace mass spectrometry	Quantifying rancidity
COLA	Beverage	Headspace mass spectrometry	Classifying soft drinks
DAIRY	Food	NIR spectroscopy	Predicting chemical composition

9 Examples: Description of Example Files

Name	Field	Data Source	Multivariate Issue
DIESEL	Automotive Fuel	Headspace mass spectrometry	Predicting composition
EUROWORK	Economics	Government statistics	Employment patterns
FUEL	Jet fuel	GC	Predicting fuel properties
HCARB3Y	Hydrocarbons	VIS/NIR spectroscopy	Predicting chemical composition
HYDROCRB	Hydrocarbons	VIS/NIR spectroscopy	Predicting chemical composition
JAVA	Beverage	Headspace mass spectrometry	Classifying coffee types
MNAPS	Environmental	Gas chromatography	Curve resolution of heavily overlapped peaks
MYCALIGN	Clinical	HPLC	Alignment of similar chromatograms
MYCOSING	Clinical	HPLC	Classifying bacterial species
OCTANE20	Gasoline	NIR spectroscopy	Predicting octane rating
OLIVEOIL	Cooking oils	Headspace mass spectrometry	Assessing adulteration
PALMS	Palm oils	Wet chemistry/GC	Classifying palm fruit type
RANDOM	Example data	Number generator	Demonstration
SEVEN	Example data	Number generator	Demonstration
TERNARY	Petrochemical	Gas chromatography	Source apportionment
XCIP4	Pharmaceuticals	NIR spectroscopy	Classifying drug excipients
XRF	Elemental	X-ray fluorescence	Predicting chemical composition

ALCOHOL.XLS

65 samples, 54 variables (53 independent, 1 class)

The file contains various blood and urine analysis results for two groups of hospital patients, those in an alcohol treatment program and those thought to be non-alcoholic. The goal is to distinguish the two groups on the basis of clinical chemistry.

ARCH.XLS

75 samples, 11 variables (10 independent, 1 class)

The file contains analysis results of obsidian samples collected from four different quarries and obsidian artifacts collected in the same region. Ten elements were determined by x-ray fluorescence. The goal is to develop a classification model from the quarry data (the first 63 samples spanning 4 quarries) and predict which quarry is the likely source for the artifacts (the last 12 samples collected in three locations). See Kowalski, et al.¹, for a detailed discussion of this data set.

BUTTER.DAT

25 samples, 108 variables (106 independent, 1 dependent, 1 class)

The file contains mass spectra collected from the headspace of butter samples that had been artificially aged to produce various levels of spoilage. The reference values for rancidity are peroxide values. The data were supplied by Leatherhead, UK.

COLA.DAT

44 samples, 107 variables (106 independent, 1 class)

This is the classic soft drink challenge: distinguishing between major brands of a soft drink in both regular and diet varieties. The data are from headspace mass spectrometry.

DAIRY.DAT

140 samples, 16 variables (14 independent, 2 dependent)

The file contains near infrared measurements on brick cheese samples at 12 wavelengths between 900 and 1100 nm. Also included as independent variables are two temperature terms (variables 13 and 14). The goal is to create a regression model that predicts fat and moisture content with the same precision as the reference wet chemical techniques.

DIESEL.DAT

44 samples, 202 variables (200 independent, 1 dependent, 1 class)

The file contains headspace mass spectra of a series of pure diesels, pure kerosenes, and mixtures of the two hydrocarbon types. The goal is to build a regression model to detect the presence and quantitate the level of kerosene added to diesel fuel.

EUROWORK.DAT

26 samples, 9 independent variables

The file contains percentages of workforce in nine different employment areas for twenty-six countries. The data were collected before the dissolution of the Soviet Union. When HCA and/or PCA are run, Eastern block countries group separately from those in the West.

FUEL.XLS

16 samples, 38 variables (35 independent, 3 dependent)

The file contains peak areas for a set of gas chromatographic runs on fuel samples. The goal is to predict physical properties of these hydrocarbon mixtures. Auto-scale the data and exclude the most intense chromatographic peak to see an improvement in prediction.

HCARB3Y.DAT

60 samples, 320 variables (316 independent, 3 dependent, 1 class)

The file contains spectra of hydrocarbon mixtures from two different diode array spectrometers. Absorbances from 470 to 1100 nm were collected. The goals are to identify spectral differences between instruments, develop a regression model to predict weight percent isooctane, toluene and decane based on samples from a single instrument, and observe the difficulties associated with transferring a calibration model to another instrument.

HYDROCRB.DAT

30 samples, 320 variables (316 independent, 5 dependent)

9 Examples: Description of Example Files

The file contains spectra of the same hydrocarbon mixtures as in HCARB3Y. The goal is to develop a regression model to predict weight percent heptane, isooc-tane, toluene, xylene and decane.

JAVA.DAT

20 samples, 121 variables (120 independent, 1 class)

The file contains the results from headspace mass spectral analysis of four different coffee roasts. The goal is to create a rapid means of assessing coffee type with-out sample preparation or special sensory expertise.

MNAPS.DAT

19 samples, 46 variables

This file contains data from a single, highly overlapped GC peak derived from an en-vironmental sample which was extracted to isolate polynuclear aromatic hydro-carbons. The goal is to resolve the peak into the underlying component profiles and spectra for subsequent quantitation.

MYCALIGN.DAT

11 samples, 1263 variables (1260 independent, 1 class, 2 dependent)

This file contains HPLC peak areas for mycolic acids in Mycobacteria cell walls. The chromatograms require alignment before subsequent classification can be done.

MYCOSING.XLS

72 samples, 39 variables (38 independent, 1 class)

This file contains HPLC peak areas for mycolic acids in Mycobacteria cell walls. The goal is to create a classification model to predict Mycobacteria species. A com-panion file SINGTEST.XLS is included to validate the model

OCTANE20.DAT

57 samples, 37 variables (36 independent, 1 dependent)

The file contains near infrared absorbances of gasoline samples taken at 20 nm in-tervals spanning 900 to 1600 nm. The goal is to develop a regression model to predict octane rating. A companion file OCT_TEST.DAT is included to validate the model

OLIVEOIL.DAT

40 samples, 102 variables (101 independent, 1 class)

This data looks at a rapid means of distinguishing olive oils. The analysis is from headspace mass spectrometry and seeks to match olive oils to their sources in different countries. As an overlay, hazelnut oil (a potential adulterant) has been added in two concentrations to some of the Greek olive oil samples.

PALMS.XLS

19 samples, 16 variables (15 independent, 1 class)

The file contains both physical property measurements and chemical compositions of a variety of tropical palms. These palms are normally classified by such visual characteristics as the size of the nut in the fruit. The goal is to determine palm type based on objective measures rather than a subjective visual method. Sev-eral sub-files of the parent PALMS file are referenced in the documentation, in-cluding PALMONES.XLS, PALMPHYS.DAT and PALMCHRO.DAT.

RANDOM.DAT

25 samples, 50 independent variables

This file contains random numbers and shows the lack of structure which appears in algorithm results produced by truly random data.

SEVEN.DAT

7 samples, 2 independent variables

This file is used in the discussions on clustering to illustrate the differences among the various linkage methods.

TERNARY.DAT

16 samples, 4232 variables (4203 independent, 3 class, 26 dependent)

This file contains the raw gas chromatograms from a set of mixtures, composed from three different oils. The chromatograms have been truncated to eliminate solvent peaks. The class variables contain the mixing proportions, while the Y variables contain marker retention times so that alignment can be run.

XCIP4.DAT

71 samples, 702 variables (700 independent, 2 class)

This file contains NIR spectra for four different common pharmaceutical excipients. One class variable distinguishes among the four compounds; the other splits one compound category by particle size. The goal is to develop a classification model to identify a sample (and its particle size where applicable) from its NIR spectrum. A companion file XCIPTTEST.DAT is included to validate the model

XRF.DAT

15 samples, 268 variables (261 independent, 7 dependent)

The file contains x-ray fluorescence spectra of nickel alloys plus elemental concentrations in the alloys as determined by wet chemistry. Four of the seven elements have specific spectral features which permit these elements to be successfully modeled using multivariate regression (i.e., PLS or PCR). See Wang, et al.², for a detailed discussion of this data set.

DATA SET REFERENCES

1. Kowalski, B.R.; Schatzki, T.F. and Stross, F.H. "Classification of Archaeological Artifacts by Applying Pattern Recognition to Trace Element Data." *Anal. Chem.* (1972) 44: 2176.
2. Wang, Y.; Zhao, X. and Kowalski, B.R. "X-Ray Fluorescence Calibration with Partial Least-Squares." *Appl. Spectrosc.* (1990) 44 (6): 998-1002,

Food and Beverage Applications

Scientists in the food and beverage industry are faced with many different quality control tasks, such as making sure that flavors meet certain standards, identifying changes in process parameters that may affect quality, detecting adulteration in ingredients and identifying the geographical origin of raw materials. Food scientists who work for regulatory agencies like the Food and Drug Administration are interested in detecting not only eco-

nomie fraud due to product substitution and adulteration but also monitoring health risks posed by food contamination.

Many of these quality control issues have traditionally been assessed by experts who evaluate product quality based on color, texture, taste, aroma, etc. Because it takes years of experience to acquire these skills, it would be advantageous to determine product quality by instrumental means.

Unfortunately, discrete sensors for qualities such as freshness or expected shelf life do not exist; therefore we must resort to measurements which, individually, may be only weakly correlated to the properties of interest. In analyzing this multivariate data, patterns emerge which are related to product quality and can be recognized by both human and computer.

THE CHEMOMETRIC APPROACH

For example, a chromatogram or spectral profile can be thought of as a fingerprint, where a pattern emerges from the relative intensities of the chromatographic sequence or spectrum. If these fingerprints are repeatable for every batch packaged for sale, it is possible for an automated quality control system to interpret those patterns in the data.

Chemometrics is a statistical approach to the interpretation of patterns in multivariate data. When used to analyze instrument data, chemometrics often results in a faster and more precise assessment of composition of a food product or even physical or sensory properties. For example, composition (fat, fiber, moisture, carbohydrate) of dairy products or grain can be quickly measured using near infrared spectroscopy and chemometrics. Food properties (*e.g.*, taste, smell, astringency) can also be monitored on a continuous basis. In all cases, the data patterns are used to develop a model with the goal of predicting quality parameters for future data.

The two general applications of chemometrics technology are:

- To predict a property of interest (typically adherence to a performance standard); and
- To classify the sample into one of several categories (*e.g.*, good versus bad, Type A versus Type B versus Type C)

SPECIFIC APPLICATIONS

This overview describes several applications in which chemometrics software has simplified methods development and automated the routine use of robust pattern matching in the food and beverage industry. The examples cited can be duplicated using Pirouette multivariate modeling software and automated in a routine quality assurance setting using either Pirouette or InStep.

Process Monitoring and Control

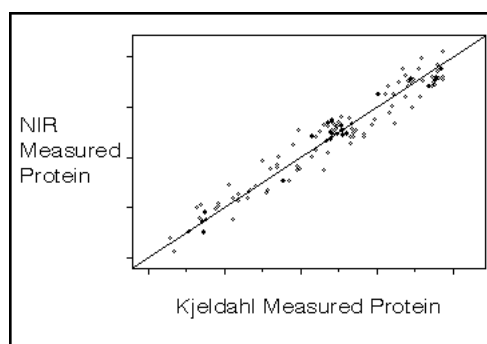
- Grading of raw materials¹
- Routine on-line quality checks²⁻³
- Minimizing sample preparation⁴
- Determining process by which product was made⁵

Much of the research and the quality control effort is aimed at assessing a product's consistency or identifying changes in process parameters that may lead to a degradation of quality standards. In most cases, no single measurement is sufficient to categorize samples for QC purposes. By examining a series of parameters simultaneously, an instru-

mental technique can be utilized that is considerably more precise than the manual spot quality checks that are the tradition. The speed and efficiency of the instrument allows chemometrics technology to be used for batch-to-batch product control⁶.

Chemometric profiling is useful for detecting changes in a process or in the ingredients; it can also be used to monitor plant-to-plant product variations. For example, near-infrared (NIR) spectroscopy can be used to determine the moisture content in packaged and prepared goods such as baking dough. NIR can also be used to monitor the carbohydrate content of grains and other natural products, which vary in composition, as they are processed. The chemometric technique has even been applied to the classification of products based on their nutritional makeup⁷.

Figure 9.1
Monitoring protein
content or other bulk
composition
properties
spectroscopically



Geographical Origin

- Identifying origin of contamination
- Determining source of ingredients by chemical composition⁸
- Tracing origin of finished products by flavor and aromatic components⁹

Chemometric pattern matching has been used in a wide variety of applications where the origin of a sample is in question. For instance, in developing a chemometric model for quality control of orange juice, two distinct groups of juice samples were shown to originate from different geographical locations (Florida and Brazil). The article demonstrated that chemical composition could be used to trace the origin of the fruit²².

Similar work in identifying a product's origin has been reported for olive oils¹⁰, brandy¹¹, wine¹² and mineral water¹³. Another study demonstrates that it is possible to relate composition patterns in wine to the wine region and the vintage year¹⁴.

Sensory Evaluation

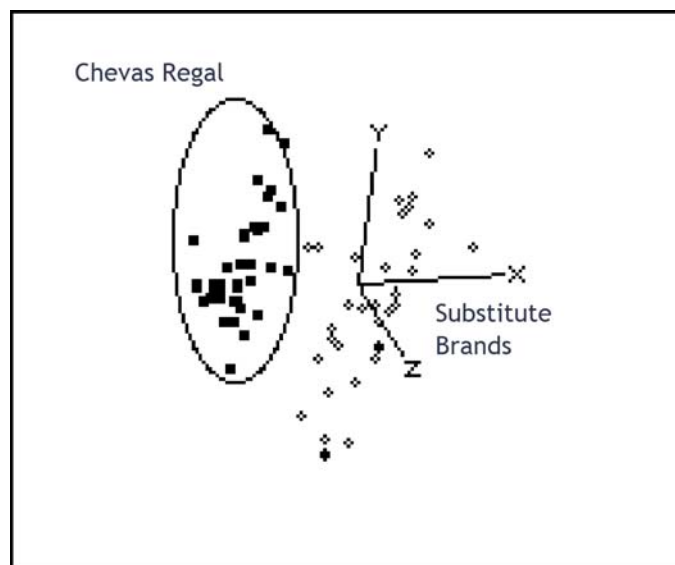
- Classification by flavor profiles¹⁵⁻¹⁶
- Replacing sensory evaluation with instrumented analysis¹⁷

A major thrust in the food and beverage industry is to bring analytical instrument techniques to play in sensory evaluation. Traditional sensory panels are expensive to maintain and can lead to inconsistent conclusions. This subjective approach to quality control can be (to some extent) replaced or enhanced by collecting chromatographic and spectroscopic information that has a high degree of correlation to sensory parameters.

Taste, smell, astringency, etc. are related to fats, oils, esters, proteins, minerals, aromatics and carbohydrates present in food. Many of these components can be profiled (finger-printed) by instrumented techniques and then correlated to sensory information by chemometric methods. The resultant statistical model can be used in on-line or routine

applications to predict flavor characteristics of unknown samples via the same instrumented technique.

Figure 9.2
Differentiating
products with trace
constituent patterns



Economic Fraud

- Identification of product adulteration, dilution and contamination¹⁸⁻²⁰
- Detection of substitution²¹

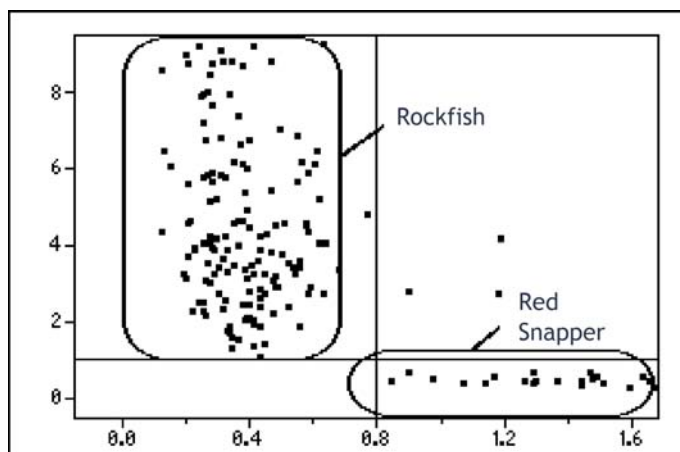
It is an unfortunate fact of life for many food producers that competitors may attempt to undercut their business by selling an adulterated product. Similarly, a less expensive, lower quality product is sometimes substituted and labelled as a more expensive product.

As an example, it has been shown that adulteration can be detected in orange juice using trace element data and chemometric techniques²². Data were collected for both orange juice and grapefruit juice, a common adulterant in “100% pure” orange juice. A chemometric model, or fingerprint, was created for each type of juice and for a blend of both juices. The model was then applied to data for new juice samples in order to determine product purity.

Another example is the unnecessary addition of water to grain. Is the amount of water added by grain resellers appropriate for dust control or is it actually economic fraud (higher weight, thus higher profit)? Monitoring the product by near infrared spectroscopy and analyzing this data with chemometrics could produce a real-time, inexpensive monitoring device²³.

Chemometrics can be used to identify instances where rockfish might be packaged and sold as red snapper. Chromatographic techniques are employed to collect data for both red snapper and rockfish; the data are then analyzed to create fingerprints for both types of fish. The model, shown in Figure 9.3, evaluates samples of fish to detect mislabeling. This system can be employed to verify that a particular chromatographic profile matches the red snapper fingerprint.

Figure 9.3
Distinguishing red
snapper from
rockfish



SUMMARY

Chemometrics is a body of statistical techniques which can correlate food quality parameters or physical properties to analytical instrument data. Patterns in the data are modeled; these models can then be routinely applied to future data in order to predict the same quality parameters. The result of the chemometrics approach is an efficiency gain in assessing product quality. The process can lead to more efficient laboratory practices or automated quality control systems. The only requirements are an appropriate instrument and software to interpret the patterns in the data.

Chemometrics software is designed to recognize patterns in virtually any type of multi-dimensional analytical data. Chemometrics can be used to speed methods development and make routine the use of statistical models for data analysis. Specifically, the application of chemometrics to the quality control of food or beverage products results in:

- More comprehensive monitoring of product quality and changes in process parameters
- Routine monitoring of raw material quality including assessment of geographical/varietal origin
- Replacement or augmentation of sensory evaluation with analytical instrument systems
- More efficient detection of product adulteration, contamination and substitution

FOOD AND BEVERAGE REFERENCES

1. Cadet, F.; Bertrand, D.; Robert, P.; Maillot, J.; Dieudonne, J. and Rouch, C. "Quantitative determination of sugar cane sucrose by multidimensional statistical analysis of their mid-infrared attenuated total reflectance spectra." *Appl. Spectrosc.* (1990) 45 (2): 166-170.
2. Lindberg, W.; Oehman, J.; Wold, S. and Martens, H. "Determination of the proteins in mixtures of meat, soymeal and rind from their chromatographic amino-acid pattern by the partial least-squares method." *Anal. Chim. Acta* (1985) 171: 1-11.
3. Robert, P.; Bertrand, D.; Devaux, M.F. and Grappin, R. "Multivariate analysis applied to near-infrared spectra of milk." *Anal. Chem.* (1987) 59 (17): 2187-2191.

4. Cowe, I.A.; Koester, S.; Paul, C.; McNicol, J. and Cuthbertson, D.C. "Principal component analysis of near infrared spectra of whole and ground oilseed rape (*Brassica napus* L.) Samples." *Chemometrics Intell. Lab. Systems* (1987) 3: 233-242.
5. Downey, G.; Robert, P.; Bertrand, D. and Kelly, P.M. "Classification of commercial skim milk powders according to heat treatment using factorial discriminant analysis of near-infrared reflectance spectra." *Appl. Spectrosc.* (1990) 44 (1): 150-155.
6. Zanini, E.; Boero, V. and Ajmone, M.F. "Effects of the elemental composition of tomatoes on their qualitative characteristics: interpretation of multivariate analysis." *Agrochimica* (1983) 27 (5-6): 373-385.
7. Vodovotz, Y.; Arteaga, G.E. and Nakai, S. "Classification of ready-to-eat breakfast cereals and food proteins using a new multivariate analysis technique." *IFT '91 Program and Exhibit Directory* (1991). *Process Monitoring and Control*
8. Stenroos, L.E. and Siebert, K.J. "Application of pattern-recognition techniques to the essential oil of hops." *J. Am. Soc. Brew. Chem.* (1984) 42 (2): 54-61.
9. Moret, I.; Scarponi, G. and Cescon, P. "Aroma components as discriminating parameters in the chemometric classification of Venetian white wines." *J. Sci. Food Agric.* (1984) 35 (9): 1004-1011.
10. Armanino, C.; Leardi, R. and Lanteri, S. "Chemometric analysis of Tuscan olive oils." *Chemometrics Intell. Lab. Systems* (1989) 343-354.
11. Miyashita, Y.; Ishikawa, M. and Sasaki, S. "Classification of brandies by pattern recognition of chemical data." *J. Sci. Food Agric.* (1989) 49: 325-333.
12. Moret, I.; Scarponi, G.; Capodaglio, G. and Cescon, P. "Characterization Soave wine by determining the aromatic composition and applying the SIMCA chemometric method." *Riv. Vitic. Enol.* (1985) 38 (4): 254-262.
13. Scarminio, I.S.; Bruns, R.E. and Zagatto, E.A.G. "Pattern recognition classification of mineral waters based on spectrochemical analysis." *Energ. Nucl. Agric.* (1982) 4 (2): 99-111.
14. Kwan, W.O. and Kowalski, B.R. "Classification of wines by applying pattern recognition to chemical composition data." *J. Food Sci.* (1978) 43: 1320-1323.
15. Van Buuren, S. "Analyzing time-intensity responses in sensory evaluation." *Food Technology* (February, 1992): 101-104.
16. Van Rooyen, P.C.; Marais, J. and Ellis, L.P. "Multivariate analysis of fermentation flavor profiles of selected South African white wines." *Dev. Food Sci.* (1985) 10 (Prog. Flavour Res.): 359-385.
17. Liu, X.; Espen, P.V. and Adams, F. "Classification of Chinese tea samples according to origin and quality by principal component techniques." *Anal. Chim. Acta* (1987) 200: 421-430.
18. Chaves das Neves, H.J. and Vasconcelos, A.M.P. "Characterization of fatty oils by pattern recognition of triglyceride profiles." *HRC & CC* (1989) 12 (4): 226-229.

19. Engman, H.; Mayfield, H.T.; Mar, T. and Bertsch, W. "Classification of bacteria by pyrolysis–capillary column gas chromatography–mass spectrometry and pattern recognition." *J. Anal. Appl. Pyrolysis* (1984) 6 (2): 137–156.
20. Headley, L.M. and Hardy, J.K. "Classification of whiskies by principal component analysis." *J. Food Sci.* (1989) 54 (5): 1351-1354.
21. Saxberg, B.E.H.; Duewer, D.L.; Booker, J.L. and Kowalski, B.R. "Pattern recognition and blind assay techniques applied to forensic separation of whiskies." *Anal. Chim. Acta* (1978) 103: 201-212.
22. Nikdel, S. and Fishback, V. "Chemometric Sleuthing: Is It Really Orange Juice" *Scientific Computing & Automation* (1989) 5(6): 19-23.
23. Devaux, M.F.; Bertrand, D.; Robert, P. and Qannari, M. "Application of multidimensional analysis to the extraction of discriminant spectral patterns from NIR spectra." *Appl. Spectrosc.* (1988) 42 (6): 1015-1019.

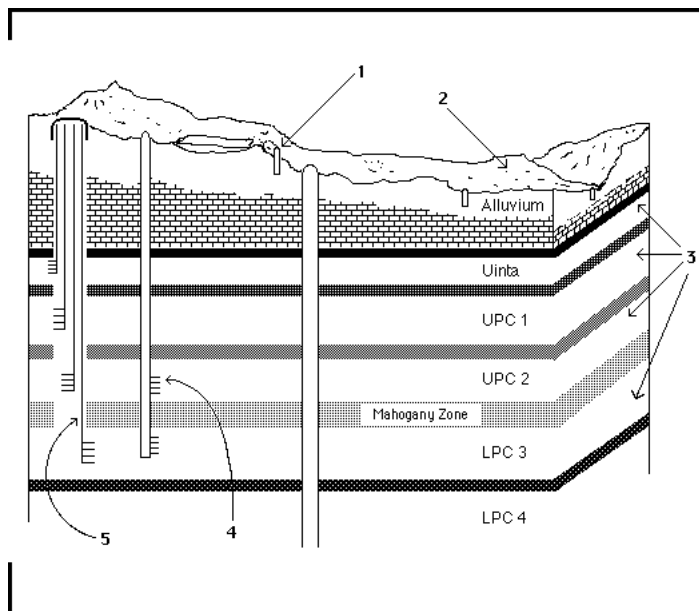
Environmental Science Applications

Scientists involved in environmental studies are faced with many different analytical tasks, such as assembling baseline studies, evaluating the contributing influence of chemical discharge to complex natural systems, and modeling biological response. Industrial scientists are concerned with the mechanics of recycling materials and maintaining process control systems that minimize pollution. Governmental control agencies, such as the EPA, are interested in detecting the presence of specific environmental agents, as well as assessing environmental damage from human sources.

Sometimes the problem is simply the enumeration of the presence or absence of constituents, whether natural or introduced. Other concerns deal with the influence that environmental factors will have on systemic response. In-field monitors and laboratory instrumentation may not directly measure these influence factors. Therefore, we are forced to make measurements of an indirect set of variables which may be only weakly correlated to the properties of interest in the system. Convenient and powerful multivariate methods have proven useful in managing and analyzing these types of complex problems.

For example, envision a chromatogram or spectral profile of a sediment extract as a fingerprint of the constituents in the sample. The pattern represents the varying amounts of the individual chemicals present. The variation contained in the signature patterns of these samples from multiple sites can reveal chemical relationships which can be characteristic of known natural phenomena or identified pollution sources.

Figure 9.4
Detecting surface discharge, characterizing an aquifer's profile and determining extent of stratigraphic leakage¹¹



Chemometrics is a multivariate mathematical and statistical approach to the analysis and interpretation of analytical data. Pattern recognition methods have been used in chemometrics to reveal and evaluate complex relationships in a wide variety of environmental applications. These methods have contributed to the systematic understanding of sediment trace metal and organic concentrations arising from natural and anthropogenic sources. Chemometrics is also useful in evaluating biological response to natural or toxic factors, and can identify the source of the contamination. Common uses of this technique are to:

- Identify factors that are combinations of measurable variables;
- Illustrate groups or cluster associations among samples;
- Assess spatial distribution of environmental factors or perturbations; and
- Predict a property of interest (such as biological response to chemical perturbation).

SPECIFIC APPLICATIONS

This overview describes a series of applications in which chemometrics software has contributed to the understanding of complex environmental systems. The examples cited can be duplicated using Pirouette and automated for routine analysis with InStep™.

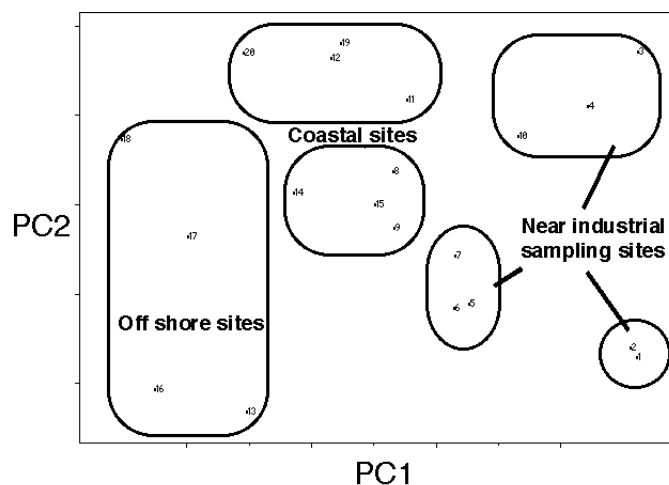
Atmospheric and Sediment Processes

- Distribution of natural or toxic chemicals¹⁻³
- Source identification, regional influence⁴⁻⁶
- Abatement and control⁷⁻⁸

Pollution modeling is generally pointed toward the identification of man-made sources of chemicals. By understanding the spatial and temporal variation of these pollutants, control measures can be applied to bring levels into compliance with environmental standards. Multivariate chemometric modeling techniques can help in situations where the emission of chemicals is added to natural sources of these same materials. The problem becomes one of discriminating the relative contributions of natural and human influence.

Chemometrics can be used to discern structure in a data set as a whole, even when individual measurements show only slight degrees of correlation. The most common use of the technology is to apportion the sources of pollution. In atmospheric studies, the relative impact of nature (such as the suspension of sea salt, or impact of forest fires) can be contrasted with suspended road dust, automotive emissions, and specific industrial contributions. Similarly, sediment studies can confirm the presence of chemicals in excess of what would be expected to occur naturally.

Figure 9.5
Classifying
environmental
samples by pollution
status and
determining
contamination
source¹⁵



The idea behind chemometric analysis is that you can effectively attribute a source to an environmental contaminant without the need to find specific marker compounds. By evaluating all of the data at once, complex data can be reduced to a set of interpretable patterns without making *a priori* assumptions about the cause of the perturbation. Not only does chemometrics supply an effective key to interpretation, the analysis yields predictive models that can be used successfully on a routine basis.

Water Management

- Pollution assessment and control⁹⁻¹¹
- Nutrient sources and dynamics¹²⁻¹³
- Trophic studies¹⁴

Inevitably, the vast majority of the waste generated by man finds its way into surface and subsurface water. Industrial and municipal effluent is pumped into bodies of water directly and the contaminants dispersed in the air or in sediments eventually is partitioned into the water table, lakes, rivers and seas. Tracking the migration of water pollution and assessing enrichment or scavenging ratios is often far more complicated than in atmospheric or sediment studies. The complication stems both from the extreme diversity of sources and from the complexity of effects as the new materials are introduced.

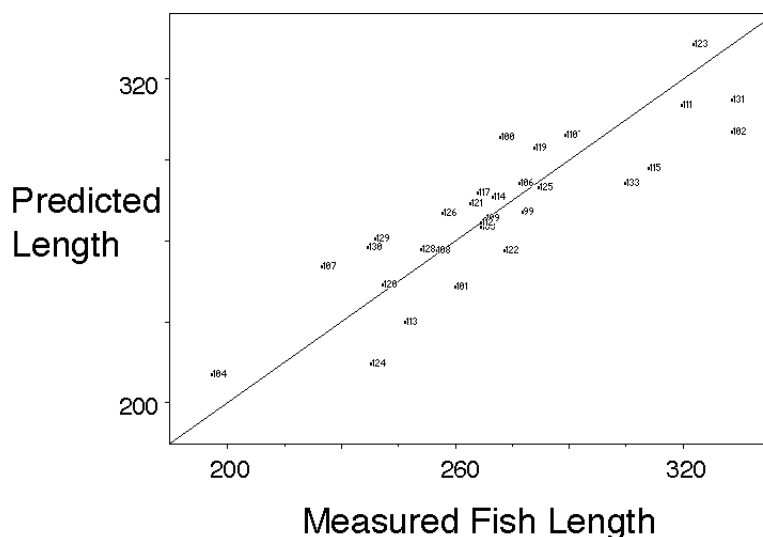
A highly useful aspect of chemometric modeling is that data used to generate patterns is not restricted to a single instrument source. Rather, the technology allows the combination of data from a variety of instrument systems as well as wet chemistry, biology and general descriptive data. An offshoot of the analysis is that, when a model is generated, you have the opportunity to assess the contributing value of individual pieces of your database relative to the inherent information content it brings to the problem. This can lead to more efficient data collection methods in future studies.

Biological Response Modeling

- Ecology and toxicity¹⁵⁻¹⁸
- Predicting species growth¹⁹
- Impact on health, tissue analysis²⁰⁻²²

While the modeling of pollution assists the evaluation and control of chemical factors in the environment, biological response modeling transforms these abstracted variables into potential health consequences. Chemometrics can allow a more efficient detailing of the influences of natural and foreign factors on the well being of specific biological systems. Furthermore, vital correlations among secondary factors may exist and be underappreciated without using this approach.

Figure 9.6
Predicting species growth patterns from lake systems dynamics and nutrient parameters



Biological response to environmental contaminants is an extremely complex process. Multivariate classification and calibration methods are particularly well-suited to extracting predictive information from a set of measurements that individually may show only a small correlation to the property of interest. Through the use of multivariate models, we can be more effective in charting relationships between species diversity or growth patterns and climatological or pollution variables. Chemometric assessments have also been used to correlate environmental factors to specific human health concerns.

Industrial Maintenance and Process Control

- Sorting of recycled material²³
- Optimizing plant effluent²⁴⁻²⁵
- Assess and control hazardous waste²⁶⁻²⁷

Industry is often portrayed as the perpetrator of pollution arising from the manufacturing process. In contrast to the popular image, most industrial plants engage in a continuous, serious effort to reduce waste (a non-economic by-product). The benefits are clear for a manufacturer to remain in compliance with regulatory agencies and to provide an increase in manufacturing efficiency.

The primary advantage of the chemometric approach in industrial settings is the relative ease of implementing a highly-focused instrument system for monitoring the quality of

a product or raw material. Most instruments sold are general purpose devices designed to generate data, but will not supply the desired information directly. A spectrophotometer can give a spectrum of a piece of plastic, but it does not specify whether it is PVC, PET, etc. Chemometrics software acts as an intermediary, interpreting the spectrum in this case, to provide the exact information desired. There is no need to build (and pay development costs for) a specific sensor system, when a general purpose instrument can be rapidly turned into a source of highly specific quality control information through a chemometric calibration process.

SUMMARY

Environmental scientists are charged with collecting and evaluating complex, inexplicit data to understand and solve concrete problems. Chemometrics is a discipline which utilizes multivariate statistical techniques, directly correlating variations in natural or toxic materials to their environmental response. Patterns in the physical and chemical data are modeled, and the models can be routinely applied to future data in order to predict comparative consequences.

Chemometrics software, such as Pirouette, is designed to recognize patterns in virtually any type of multidimensional analytical data. Chemometrics can be used to speed methods development and make routine the use of statistical models for data analysis. Specifically, the application of chemometrics to environmental analysis can result in:

- Detection of pollution contributions from a complex mixture of sources;
- Assessment of geographical or atmospheric distributions and influences;
- Prediction of biological response to perturbation;
- Understanding the interplay of influential factors which cannot be directly measured;
- Optimization of processes for controlling plant waste or recycling; and
- Improvement in the interpretability of analytical instrument data.

SELECTED ENVIRONMENTAL REFERENCES

1. Gomez, M.L.S. and Martin, M.C.R. "Application of cluster analysis to identify sources of airborne particles" *Atmos. Environ.* (1987) 21 (7): 1521-1527.
2. Hooper, R.P. and Peters, N.E. "Use of multivariate analysis for determining sources of solutes found in wet atmospheric deposition in the United States." *Environ. Sci. Technol.* (1989) 23:1263-1265.
3. Wenning, R.J.; Harris, M.A.; Unga, M.J.; Paustenbach, D.J. and Bedbury, H. "Chemometric comparisons of polychlorinated dibenzo-p-dioxin and dibenzofuran residues in surficial sediments from Newark Bay, New Jersey and other industrialized waterways." *Arch. Environ. Contam. Toxicol.* (1992) 22: 397-413.
4. Boni, C.; Caruso, E.; Cereda, E.; Lombardo, G.; Braga Marcazzan, G.M. and Redaelli, P., "Particulate matter elemental characterization in urban areas: pollution and source identification." *J. Aerosol Sci.* (1988) 19 (7):1271-1274.
5. Cohen, M.A.; Ryan, P.B.; Spengler, J.D.; Ozkaynak, H. and Hayes, C. "Source-receptor study of volatile organic compounds and particulate matter in the Kanawha Valley, WV - II. analysis of factors contributing to VOC and particle exposures." *Atmos. Environ.* (1991) 25B (1): 95-107.

6. Winters, G.V. and Buckley, D.E., "Factor analyses as a method of evaluating spatial and temporal variations in sediment environmental quality in Halifax Harbour." Pittsburgh Conference Paper #93-480 (1993) Atlanta, GA.
7. Kim, D.S.; Hopke, P.K.; Massart, D.L.; Kaufman, L. and Casuccio, G.S., "Multivariate analysis of CCSEM auto emission data." *Sci. Total Environ.* (1987) 59:141-155.
8. Vong, R.; Geladi, P.; Wold, S. and Esbensen, K. "Source contributions to ambient aerosol calculated by discriminant partial least squares regression." *J. Chemometrics* (1988) 2: 281-296.
9. Bomboi, M.T.; Hernandez, A.; Marino, F. and Hontoria, E. "Application of multivariate analysis for characterization of organic compounds from urban runoff." *Sci. Total Environ.* (1990) 93: 523-536.
10. Higashi, K. and Hagiwara, K. "Identification of oil spilled at sea by high performance gel permeation chromatography pattern recognition" *Wat. Sci. Tech.* (1988) 20 (67): 55-62.
11. Meglen, R.R. and Erickson, G.A. "Application of pattern recognition to the evaluation of contamination from oil shale retorting" in Environment and Solid Waste, Characterization, Treatment and Disposal C.W. Francis and S.I. Auerbach eds., Butterworth Publishers, Boston (1983): pp369-381.
12. Grimalt, J.O.; Olive, J. and Gomez-Belinchon, J.I. "Assessment of organic source contributions in coastal waters by principal component and factor analysis of the dissolved and particulate hydrocarbon and fatty acid contents" *Intern. J. Environ. Anal. Chem.* (1990) 38: 305-320.
13. Vong, R.J.; Hansson, H.; Ross, H.B.; Covert, D.S. and Charlson, R.J. "Northeastern Pacific submicrometer aerosol and rainwater composition: a multivariate analysis." *J. Geophys. Res.* (1988) 93 (D2):1625-1637.
14. Lamparczyk, H.; Ochocka, R.J.; Grzybowski, J.; Halkiewicz, J. and Radecki, A. "Classification of marine environment samples based on chromatographic analysis of hydrocarbons and principal component analysis" *Oil & Chem. Pollution* (1990) 6:177-193.
15. Macdonald, C.R.; Norstrom, R.J. and Turle, R. "Application of pattern recognition techniques to assessment of biomagnification and sources of polychlorinated multicomponent pollutants, such as PCBs, PCDDs and PCDFs" *Chemosphere* (1992) 25 (1-2):129-134.
16. Simmler, N. and Schulten, H. "Pattern recognition of spruce trees: an integrated, analytical approach to forest damage" *Environ. Sci. Technol.* (1989) 23:1000-1006.
17. Stalling, D.L.; Norstrom, R.J.; Smith, L.M. and Simon, M. "Patterns of PCDD, PCDF, and PCB contamination in Great Lakes fish and birds and their characterization by principal components analysis" *Chemosphere* (1985) 14 (6-7): 627-643.
18. Von Rudloff, E.; Lapp, M.S. and Yeh, F.C. "Chemosystematic study of *Thuja plicata*: multivariate analysis of leaf oil terpene composition", *Biochemical Systematics and Ecology* (1988)16 (2):119-125.

19. Moseholm, L. "Analysis of air pollution plant exposure data: the soft independent modeling of class analogy (SIMCA) and partial least squares modeling with latent variable (PLS) approaches", *Environ. Pollution* (1988) 53: 313-331.
20. Brakstad, F. "A comprehensive pollution survey of polychlorinated dibenzo-p-dioxins and dibenzofurans by means of principal component analysis and partial least squares regression" *Chemosphere* (1992) 24 (12):1885-1903.
21. Diaz-Caneja, N.; Gutierrez, I.; Martinez, A.; Matorras, P. and Villar, E. "Multivariate analysis of the relationship between meteorological and pollutant variables and the number of hospital admissions due to cardio-respiratory diseases" *Environ. International* (1991) 17: 397-403.
22. Vogt, N.B. "Polynomial principal component regression: an approach to analysis and interpretation of complex mixture relationships in multivariate environmental data" *Chem. Intell. Lab. Sys.* (1989) 7: 119-130.
23. Alam, M.K. and Stanton, S.L. "Sorting of waste plastics using near-infrared spectroscopy and neural networks" *Process Control and Quality* (1993) 4: 245-257.
24. Brown, S.D.; Skogerboe, R.K. and Kowalski, B.R., "Pattern recognition assessment of water quality data: coal strip mine drainage" *Chemosphere* (1980) 9: 265-276.
25. Gonzalez, M.J.; Fernandez, M.; Jimenez, B. and Hernandez, L.M. "Application of Multivariate Analysis to the Determination of PCBs in Fly Ash From Municipal Incinerators" *Chemosphere* (1989) 18 (11/12): 2213-2227.
26. O'Brien, R.; Sinha, B.K. and Smith W.P. "A Statistical Procedure to Evaluate Clean-up Standards" *J. Chemometrics* (1990) 5 (3): 249-261.
27. Sarker, M.; Glen, W.G.; Yin, L.; Dunn III, W.J.; Scott, D.R. and Swanson, S. "Comparison of simca pattern recognition and library search identification of hazardous compounds from mass spectra" *Anal. Chim. Acta* (1990) 257: 229-238.

Chemometrics in Chromatography

Chromatography is an extremely versatile technique for the analytical laboratory. The chromatographic patterns generated by modern instruments are used in a wide variety of quantitative and qualitative analyses. The techniques are robust enough (and we have assembled experience enough) to allow a rapid development of chromatographic methods and move this experience into routine use in an analytical laboratory, quality control laboratory, or even an in-line process setting.

At least three goals can be identified for projects which use chromatographic instrumentation:

- Quantitation of the components in an analysis mixture
- Separation of components in the mixture for purposes of fraction collection
- Matching of the chromatographic patterns to an experience set or library

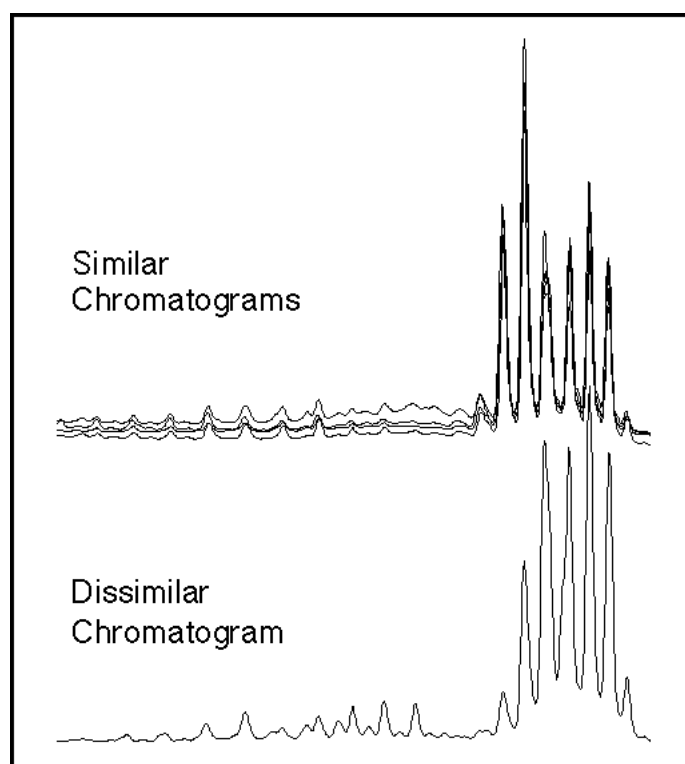
Although it is not always an expressed goal of a chromatographic analysis, we commonly use human pattern recognition skills to interpret the instrument output. The purpose of this pattern recognition step is usually to classify the sample in some way (*e.g.*, is the

sample of acceptable quality or is the sample consistent with a previous run?). Through the methods development process, we often strive to develop a set of rules-of-thumb for interpreting patterns. Often these heuristics involve calculating the ratio of two intensities or developing a simple decision tree based on a series of features in the chromatographic trace.

This overview describes a series of applications in which pattern recognition software has simplified methods development and automated the routine use of robust pattern matching in chromatography. The field of study which encompasses this technology is called chemometrics and the examples cited can be duplicated using Pirouette multivariate modeling software.

A chromatogram can be thought of as a chemical fingerprint where the pattern emerges from the relative intensities of the sequence of peaks passing by the detector.

Figure 9.7
Classification of chromatograms is based on the relative abundance of all the peaks in the mixture



Chromatographic fingerprinting, whether by human intervention or automated in software, is used in two generic application areas:

- To infer a property of interest (typically adherence to a performance standard); or
- To classify the sample into one of several categories (good versus bad, Type A versus Type B versus Type C, etc.).

The following sections contain examples of the use of chemometric technology to problems in chromatographic pattern recognition, with applications drawn from different industries.

SPECIFIC APPLICATIONS

Pharmaceutical/Biotech

- Protein mapping for product quality control
- Grading of raw materials
- Drug identification¹

Much of the research and the quality control effort is aimed at assessing a product's consistency or identifying changes in process parameters that may lead to a degradation of quality standards. In most cases, no single concentration is sufficient to categorize samples for QC purposes. As newly bioengineered forms of products make their way to the market, the lack of standards will drive a further need for pattern recognition technology for batch-to-batch product control.

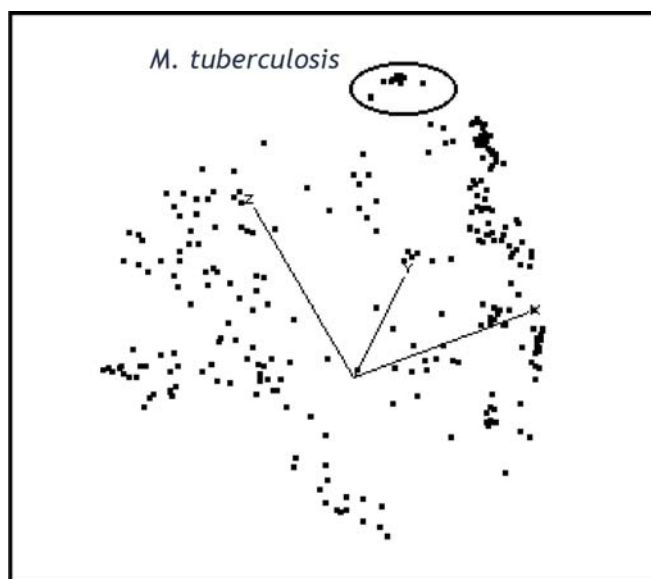
Medical/Clinical

- Identification of microbial species by evaluation of cell wall material²⁻³
- Cancer profiling and classification
- Predicting disease state⁴⁻⁶

A prime concern of clinical diagnosis is to classify disorders rapidly and accurately. Chemometric techniques can be applied to chromatographic data to develop models allowing clinicians to distinguish among disease states based on the patterns in body fluids or cellular material.

All living systems consist of chemical compounds and the relative distribution of these constituents can be used as a biological fingerprint to type samples. Bacteria, yeast and molds are commonly classified using matching techniques on chromatographic patterns. One example is the identification of the organism causing tuberculosis and related mycobacterial species using HPLC.

Figure 9.8
M. tuberculosis can be identified by examining mycolic acid distribution in bacterial cell walls. This figure shows a 3D representation of samples of more than 30 species.



Food/Beverage

- Replacing sensory evaluation with instrumented analysis
- Geographical/variety origin
- Competitor evaluation (change in process, constituents)
- Beer, wine quality control, classification⁷⁻¹⁰
- Proving economic fraud¹¹

A constant issue in the food industry is the analysis of raw materials and finished products to insure consistency and quality. Chromatographic profiling is useful in detecting changes in a process or in the ingredients and can also be used to monitor plant-to-plant product variations.

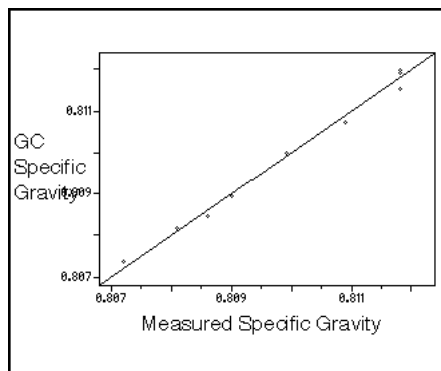
A second thrust in the food and beverage industry is to bring analytical instrument techniques to play in sensory evaluation. Traditional sensory panels are expensive to maintain and can lead to inconsistent conclusions. This subjective approach to quality control can be (to some extent) replaced or enhanced by adding the more objective chromatography/chemometrics technique.

Chemical/Petroleum

- Oil exploration (oil-oil correlation, oil-source rock correlation)¹²
- Refinery QC (product uniformity, raw material variation)

Organic geochemistry often involves the chromatographic analysis of hydrocarbon extracts from geologic formations or oil samples. The patterns reflected in the chromatograms are a combination of biological origin and any geologic alteration. Interpretation of the chromatographic traces can be automated using chemometrics.

Figure 9.9
Physical properties, such as the specific gravity of jet fuel, can be determined via calibration of the GC trace to a density measurement



Environmental

- Evaluation of trace organics and pollutants¹³⁻¹⁴
- Pollution monitoring where multiple sources are present
- Effective extraction of information from large environmental databases

Environmental studies constitute a large portion of the research and monitoring money spent in the world today. This expenditure reflects the concern for the effect chemicals have on the health of the earth's eco system. A typical data set involves the collection of a large amount of data from a diverse set of instrument sources. Chromatography plays

a central role in this data assembly because of its sensitivity and specificity for many of the organic compounds of interest.

Chemometric techniques provide the means to extract usable information from the environmental measurements. Through these pattern recognition and modeling techniques, improved descriptions of pollution patterns and their sources are available to the analyst¹⁵.

Forensics

- DNA fingerprinting
- Arson investigation
- Geographical origin of illegal substances

In forensic analysis, the issue is not to determine the concentration of various chemical constituents, but rather to determine if a chromatographic trace is correlated to a known sample. Chemometric pattern matching has been used in a wide variety of applications where the origin of a sample is in question.

SUMMARY

Chemometrics can be used to accomplish a variety of goals in the chromatography laboratory:

- Speeding of methods development
- More effective multivariate calibration
- Detection and monitoring of impurities

Today's chromatographers have jobs to do that extend beyond the act of collecting, analyzing and harvesting reports on individual samples. The true product of the analytical endeavor lies in the consolidation of these individual analyses into an evaluation of the chemical system as a whole. We compare a new sample against a compilation of our prior experience, we try to infer properties of interest with non-specific analytical tools, etc.

Chemometrics can be used to condense large assembly projects into more manageable time frames; the modeling capability allows you to speed methods development and interpretation of complex chromatographic patterns. The multivariate models can be placed in an expert-system context to allow robust implementation of very customized chromatographic systems¹⁶⁻¹⁹.

Pirouette is designed to recognize patterns in virtually any type of analytical data. The process can be used to speed methods development and make routine the use of multivariate statistical models. The examples described in this note are easily duplicated or can be used as analogies for custom analyses.

SELECTED CHROMATOGRAPHY REFERENCES

1. Musumarra, G.; Scarlata, G.; Romano, G.; Cappello, G.; Clementi, S. and Giulietti, G. "Qualitative Organic Analysis. Part 2. Identification of Drugs by Principal Components Analysis of Standardized TLC Data in Four Eluent Systems and of Retention Indices on SE 30." *J. Anal. Toxicology* (1987) 11 (Jul./Aug.): 154–163.
2. Engman, H.; Mayfield, H.T.; Mar, T. and Bertsch, W. "Classification of bacteria by pyrolysis–capillary column gas chromatography–mass spectrometry and pattern recognition." *J. Anal. Appl. Pyrolysis* (1984) 6 (2): 137–156.

3. Butler, W.R.; Jost, K.C. and Kilburn, J.O. "Identification of mycobacteria by high performance liquid chromatography." *J. Clin. Microbiol.* (1991) 29 (11): 2468-2472.
4. Kowalski, B.R. "Measurement analysis by pattern recognition" *Anal. Chem.* (1975) 47:1152A-
5. Marshall, R.J.; Turner, R.; Yu, H. and Cooper, E.H. "Cluster analysis of chromatographic profiles of urine proteins." *J. Chromatogr.* (1984) 297: 235–244.
6. Pino, J.A.; McMurry, J.E.; Jurs, P.C. and Lavine, B.K. "Application of pyrolysis/gas chromatography/pattern recognition to the detection of cystic fibrosis heterozygotes." *Anal. Chem.* (1985) 57 (1): 295–302.
7. Moret, I.; Scarponi, G. and Cescon, P. "Aroma components as discriminating parameters in the chemometric classification of Venetian white wines." *J. Sci. Food Agric.* (1984) 35 (9): 1004–1011.
8. Moret, I.; Scarponi, G.; Capodaglio, G. and Cescon, P. "Characterization Soave wine by determining the aromatic composition and applying the SIMCA chemometric method." *Riv. Vitic. Enol.* (1985) 38 (4): 254–262.
9. Stenroos, L.E. and Siebert, K.J. "Application of pattern–recognition techniques to the essential oil of hops." *J. Am. Soc. Brew. Chem.* (1984) 42 (2): 54–61.
10. Van Rooyen, P.C.; Marais, J. and Ellis, L.P. "Multivariate analysis of fermentation flavor profiles of selected South African white wines." *Dev. Food Sci.* (1985) 10 (Prog. Flavour Res.): 359–385.
11. Saxberg, B.E.H.; Duewer, D.L.; Booker, J.L. and Kowalski, B.R. "Pattern recognition and blind assay techniques applied to forensic separation of whiskies." *Anal. Chim. Acta* (1978) 103: 201-212.
12. Zumberge, J.E. "Prediction of source rock characteristics based on terpane biomarkers in crude oils: A multivariate statistical approach." *Geochim. Cosmochim. Acta* (1987) 51 (6): 1625-1637.
13. Dunn, W.J.; Stalling, D.L.; Schwartz, T.R.; Hogan, J.W.; Petty, J.D.; Johansson, E. and Wold, S. "Pattern recognition for classification and determination of polychlorinated biphenyls in environmental samples." *Anal. Chem.* (1984) 56 (8): 1308–1313.
14. Onuska, F.I.; Mudroch, A. and Davies, S. "Application of chemometrics in homolog–specific analysis of PCBs." *HRC & CC* (1985) 8: 747–754.
15. Breen, J.J. and Robinson, P.E., Eds., Environmental Applications of Chemometrics ACS Symposium Series (1985) 292: 286pp.
16. Chien, M. "Analysis of complex mixtures by gas chromatography/mass spectrometry using a pattern recognition method." *Anal. Chem.* (1985) 57 (1): 348–352.
17. Isaszegi–Vass, I.; Fuhrmann, G.; Horvath, C.; Pungor, E. and Veress, G.E. "Application of pattern recognition in chromatography." *Anal. Chem. Symp. Ser.* (1984) 18 (Mod. Trends Anal. Chem., Pt. B): 109–124.

18. Smith, A.B.,; Belcher, A.M.; Epple, G.; Jurs, P.C. and Lavine, B. "Computerized pattern recognition: a new technique for the analysis of chemical communication." *Science* (1985) 228 (4696): 175–177.
19. Stepanenko, V.E. "Group analysis and pattern recognition as a basis for chromatographic identification." *Zh. Anal. Khim.* (1985) 40 (5): 881–886.

Part III.

Software Reference

10 The Pirouette Interface

11 Object Management

12 Charts

13 Tables

14 Data Input

15 Output of Results

16 Pirouette Reference

The Pirouette Interface

Contents

Overview	10-1
Ribbon Buttons	10-3
Cursors	10-6
View Preferences	10-7
Chart Preferences	10-16
Other Preferences	10-19
Preference Sets	10-21

This chapter addresses the more general aspects of the Pirouette graphics interface. It describes the various ribbon buttons and cursor shapes and explains how to modify Pirouette's appearance through the setting of preferences. For detailed descriptions of how to work with graphics and spreadsheet views, consult [Chapter 12, Charts](#) and [Chapter 13, Tables](#), respectively.

Overview

Graphic displays in Pirouette are not static—you can interact with any graphic and obtain additional information from it. Interaction is accomplished via a set of tools which often require selecting an item. This is accomplished by clicking on the item with the left mouse button, an action sometimes described as *highlighting* because the appearance of the selected item changes to indicate its altered state. Often it is necessary to select multiple items. Selecting conventions for lists and graphics are explained in the next two sections.

SELECTING IN LISTS AND TABLES

For multiple, contiguous selections from tables and lists, there are two approaches: click-drag and Shift-select.

- To click-drag, click down and drag the mouse cursor, let the list scroll to the desired item, then release the mouse.
- To Shift-select, click on the first item, then, after scrolling down, click on the last desired item with the Shift key held down.

To make discontinuous selections, use the Ctrl-select approach:

- With the Ctrl key held down, click on additional items. Ctrl-select can be repeated as often as necessary; the selections will accumulate.

SELECTING IN GRAPHICS

Selecting points in the dendrogram and in 2D and 3D plots using the Pointer tool is similar to the process outlined above.

- To select a point or group of points, place the cursor in an allowed region of the graphic (where the cursor is displayed in its Pointer form), then click–drag diagonally to draw a rubber box. All points (or leaves, in the case of a dendrogram) within the box become highlighted when the mouse button is released.
- To reverse the selection state a group of points, Ctrl–click–drag around them. When the mouse button is released, previously highlighted points become unhighlighted and previously unhighlighted points become highlighted. The selection state of points outside the box is not affected.
- To add more points to the selection, Shift–click–drag around them. Previously highlighted points remain highlighted and the newly selected points also become highlighted.
- To deselect all points, click once in an allowed region with the mouse button. All selected points lose their highlighting.

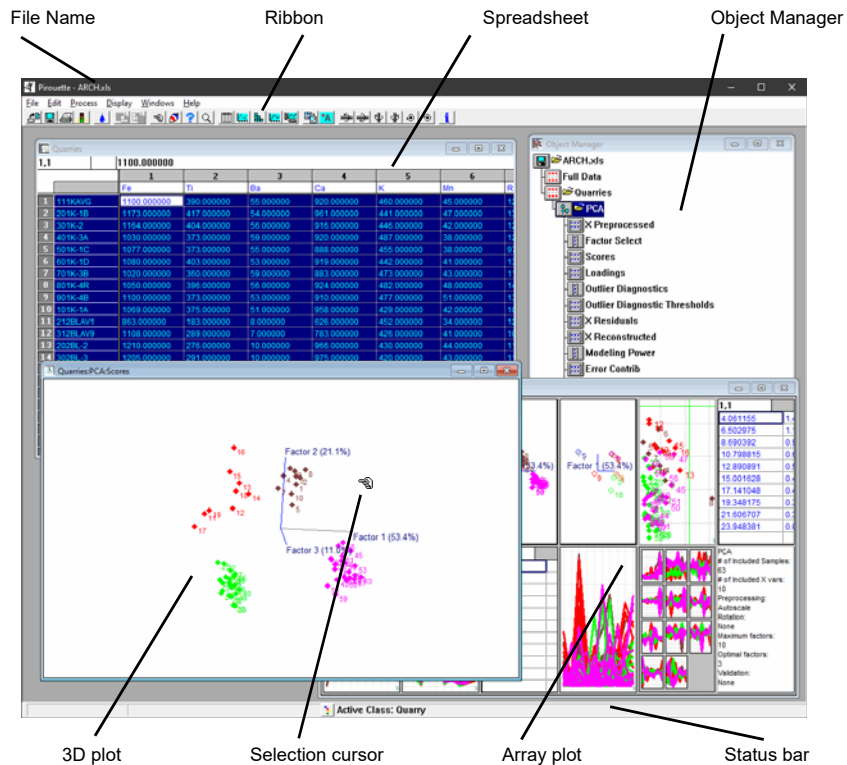
Selecting points in line plots using the Pointer tool is also similar.

- To select a single line, click on it. The line will change from thin to thick to indicate its highlighted state.
- To select multiple lines, click–drag the cursor such that the rubber box overlaps a set of lines, and those lines will become highlighted.
- To reverse the state of a set of lines, Ctrl–click–drag around them. Previously highlighted lines become unhighlighted and previously unhighlighted lines become highlighted.
- To add more lines to the selection, Shift–ctrl–drag around them. Previously highlighted lines remain highlighted and the newly selected lines become highlighted.

THE PIROQUETTE WINDOW

The Pirouette environment has many structures common to Windows programs as well as a few unique features. The following figure illustrates some of these features. Some are important enough to warrant their own chapters: “[Object Management](#)”, “[Tables](#)” and “[Charts](#)” in particular. Refer to the appropriate chapter for more detailed discussion.

Figure 10.1
Features of the
Pirouette
environment








Ribbon Buttons

The button ribbon provides easy mouse access to common program features. It is always located along the top of the Pirouette window, as shown in Figure 10.1. The suite of buttons displayed at any time is a function of the current window type (Object Manager vs. Chart). Displayed buttons are normally enabled; but if an action is not available, the button is grayed. The buttons are divided into groups. The tables which follow contain references to button function.

FILE AND PROCESSING FUNCTIONS

Common file-oriented actions comprise the core ribbon set; these buttons, shown below, are present regardless of what view is current.




Table 10.1
Ribbon buttons for
file and processing
functions

Button	Description	Reference
	Load a data file from disk	“Open Data” on page 16-4
	Save a data file to disk	“Saving Data” on page 15-4
	Print current chart view	“Print” on page 16-9
	Configure and run an algorithm	“Run” on page 16-20
	Open Pirouette Help	“Setup” on page 16-45

WINDOW MANIPULATIONS

Three buttons are shown when charts are displayed; they manipulate windows.






Table 10.2
Ribbon buttons for window manipulations

Button	Description	Reference
	Activate the Grabber for drag and drop	"Creating Charts with the Drop Button" on page 12-3
	Contract (that is, unzoom) a zoomed subplot back to its originating array	"Multiplots" on page 12-20
	Zoom a subplot to the full window	"Multiplots" on page 12-20

INTERACTION TOOLS

Pirouette's interaction tools manipulate its graphics. When an interaction tool is selected by clicking on its button, the cursor changes to a form specific to that tool.




Table 10.3
Ribbon buttons for interaction tools

Button	Description	Reference
	Select points in 2D or 3D plot	"Selecting Points" on page 12-5
	Rotate a 3D plot	"Spinning a 3D Plot" on page 12-9
	Identify points or lines	"Identifying Points" on page 12-7 and "Identifying Lines" on page 12-15
	Magnify plot regions	"Magnifying Regions" on page 12-8
	Select ranges in line plots	"Selecting Ranges" on page 12-18

EDITING

Common spreadsheet editing functions are assigned ribbon buttons.


Table 10.4
Ribbon buttons for editing






Button	Description	Reference
	Cut	"Cut" on page 16-13
	Copy	"Copy" on page 16-13
	Paste	"Paste" on page 16-14

VIEW SWITCHING

The six buttons shown below switch the current graph window from one view to another. Clicking on one of these buttons causes the current window to immediately update with the new view type.

Table 10.5
Ribbon buttons for view switching





Button	Description	Reference
	Table	Chapter 13, Tables

Button	Description	Reference
	3D plot	“Scatter Plots” on page 12-5
	2D plot	“Scatter Plots” on page 12-5
	Line plot	“Line Plots” on page 12-13
	Line plot for factor selection	“Factor Selection Line Plots” on page 12-19
	Multiplot	“Multiplots” on page 12-20

PLOT CUSTOMIZATION

The buttons listed in below customize graphics.






Table 10.6
Ribbon buttons for plot customization

Button	Description	Reference
	Cloaking	“Cloaking” on page 12-8
	Label	“Point Labels” on page 12-7
	Selector	“Specifying Axes” on page 12-5 and “Specifying Axes and Orientation” on page 12-14
	Redraw	“Redrawing Traces” on page 12-19

NAVIGATION AIDS

The ribbon includes navigation shortcuts for both the Object Manager and for table views of raw data.

Table 10.7
Ribbon buttons for navigation aids

Button	Description	Reference
	Contract Object Manager tree	“Navigation” on page 11-2
	Expand Object Manager tree	“Navigation” on page 11-2
	Jump to X block	“Variable Type Blocks” on page 13-5
	Jump to C block	“Variable Type Blocks” on page 13-5
	Jump to Y block	“Variable Type Blocks” on page 13-5















SPINNER CONTROL

The final six buttons rotate axes in 3D plots; see [“Spinning with Spin Control Buttons” on page 12-10](#) for a description.

Cursors

In Pirouette, the cursor's appearance hints at what type of interaction is possible. The following table summarizes the cursors and indicates which are controlled by the Windows operating system.

Table 10.8
Pirouette cursors

Cursor	Description
	Arrow (system)
	I-beam insertion cursor (system)
	Hourglass (system)
	Plus, tables only
	Double-headed arrow, dendrogram only
	Single-headed arrow, dendrogram only
	Pointer
	Spinner, 3D plot only
	ID
	Magnifier
	Range selector, line plot only
	Grab
	Drag
	Drop

The arrow cursor selects windows, items from menus, objects in dialog boxes and ribbon functions. It is also used to navigate the dendrogram as described in “[Dendrogram Navigation](#)” on page 12-25. The cursor takes on an hourglass form whenever the system is busy and all other operations are suspended.

A cursor changes to the insertion shape when it passes over an Edit Field in the spreadsheet or in a dialog box. When the cursor is not over the Edit Field in the spreadsheet, the plus form is displayed. If the mouse button is clicked while the plus is over a cell, the cell under the cursor is selected and its contents appear in the Edit Field. If the plus is over a column/row index when clicked, the entire column/row is selected.

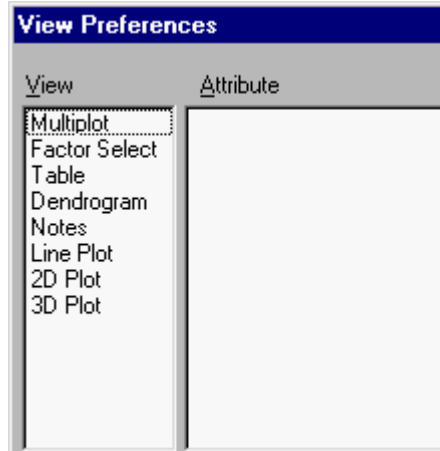
The horizontal double-headed arrow and a single arrow pointing right both appear in the dendrogram; see “[Dendrogram Navigation](#)” on page 12-25 for a discussion of their functionality. The pointer, spinner, ID, magnifier, and range selection cursors appear in graphics windows active when the particular interaction tool was selected; they are discussed in “[Interaction Tools](#)” on page 10-4.

Finally, the last three entries in [Table 10.8](#) appear when a drag and drop is initiated, continued and completed, respectively; see “[Creating Charts](#)” on page 12-1 for a description of these actions.

View Preferences

Often users want to customize colors and text attributes for the various views in Pirouette. Choosing Preferences > View on the Windows menu opens the dialog box shown in the following figure.

Figure 10.2
List of views with preferences



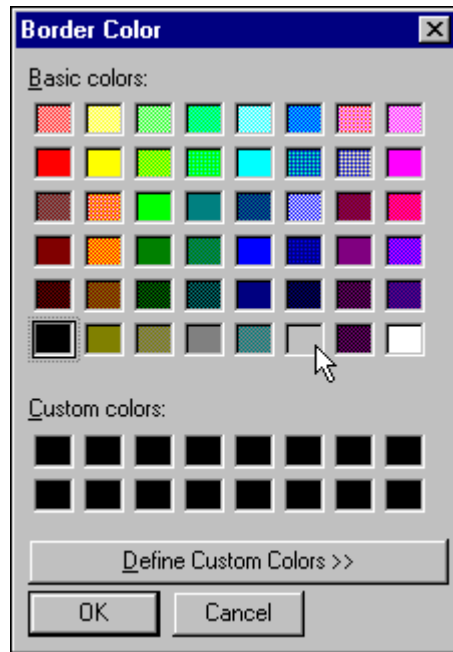
To change a view attribute:

- Click on a View type and inspect its Preview
- Double-click on the Attribute to modify
- Change Attribute parameters (color, font, etc.)
- Click on OK
- Check the Preview to determine if the changes are satisfactory
- Continue making changes and then click on OK to exit Preferences

COLOR ATTRIBUTES

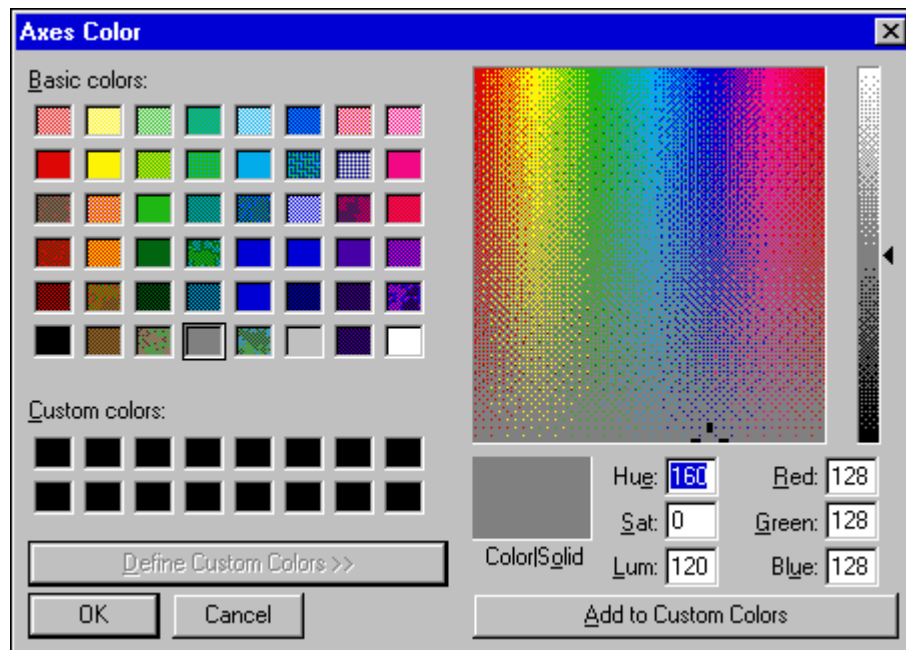
Attributes with *color* in their name are modified via the Windows Color Picker shown below. Initially, the palette of standard colors is displayed with the current color block marked with a thick border.

Figure 10.3
Color picker dialog



To create a new color, click on the Define Custom Colors button and the window expands as shown in Figure 10.4. Consult your Windows documentation for more details on the color picker.

Figure 10.4
Defining a custom
color

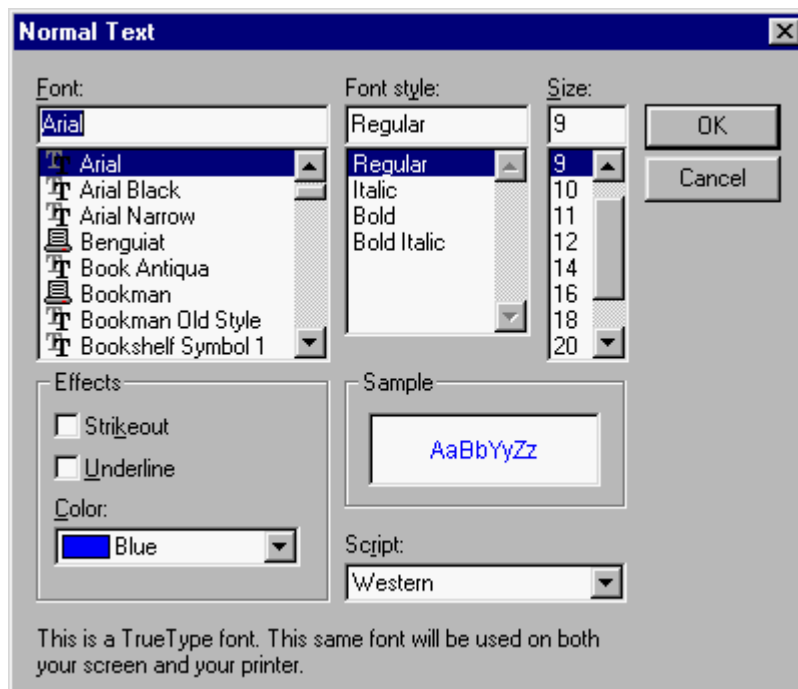


TEXT ATTRIBUTES

Attributes with *text* in their name are modified via a dialog box like that shown in Figure 10.5. In most cases, you can set the font, style and size. In addition, some Pirouette

plot and table features can have a distinct text color. These colors are chosen from a list in this same Text dialog box.

Figure 10.5
Text attributes dialog



GRID

Grid lines can be displayed on 2D and line plots. These can be helpful to verify values of plotted points but can also distract when trying to observe relationships among many points. To show or hide the grid lines, double-click on the Grid attribute for the desired view and make your selection.

Figure 10.6
Grid dialog



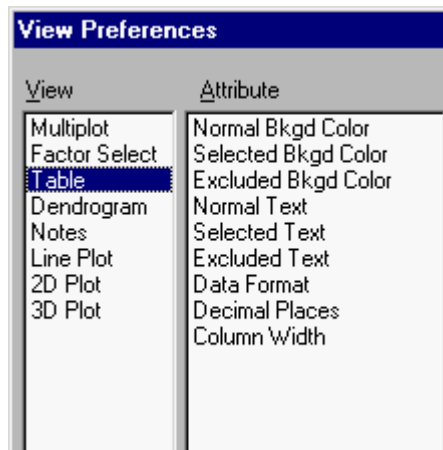
OTHER ATTRIBUTES

In this section, view type attributes are briefly described if they are not self-explanatory. Attributes associated with more than one view type are described only once.

Table

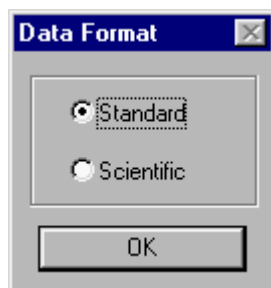
The first three attributes modifiable in table views are the background colors of the text fields in a table: normal, selected (highlighted), and for excluded columns or rows. The next three attributes control the text in these fields and can also be modified.

Figure 10.7
Table view attributes



The Data Format attribute toggles between normal or scientific notation.

Figure 10.8
Data Format dialog



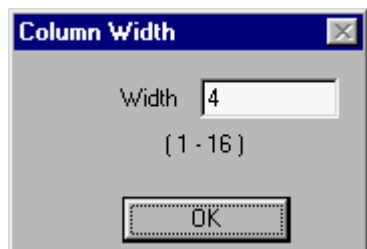
You can specify the number of decimal places in numbers in tables:

Figure 10.9
Decimal Places
dialog



Use the dialog box shown in the following figure to set column width.

Figure 10.10
Column Width dialog

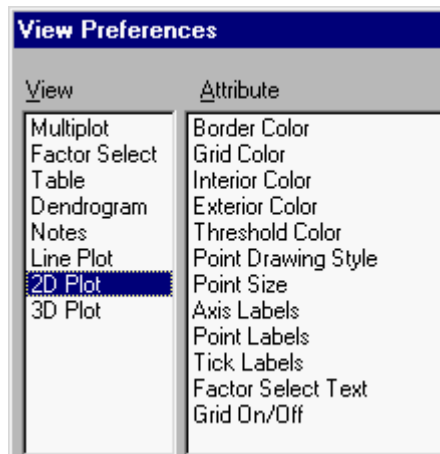


The actual number of characters displayed in a column depends on font (proportional or monospaced), number of decimal places, and column width.

Scatter Plots

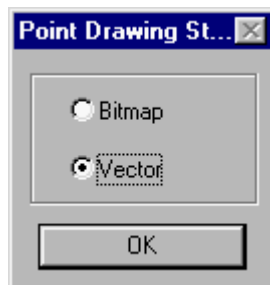
Scatter plots have several color and text attributes. Each text attribute also controls color, except for Point Labels. The color of points and their labels is determined by the color map as explained in “Color Sequence” on page 10-18. Select the color attributes for the border, grid, interior, exterior, and the reference line via a dialog box like that in Figure 10.3. Label attributes are set via a dialog box like that in Figure 10.5 as is the font for the optimal factor value shown in the plot space.

Figure 10.11
2D view attributes



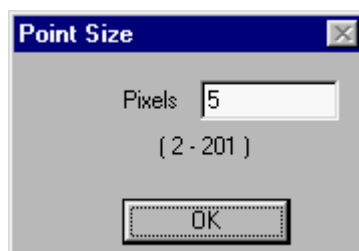
The appearance of plot symbols in a scatter plot is determined by the Point Style and Point Size attributes.

Figure 10.12
Point Style dialog



Using a vector-based point symbol may improve print quality as drawings based on vectors display well at any size, unlike bitmaps which print well only at multiples of their original resolution.

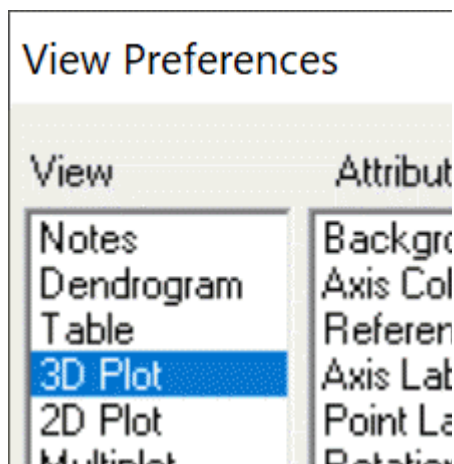
Figure 10.13
Point Size dialog



Choosing a larger point symbol may improve visibility, particularly at higher resolution, but points may appear more overlapped than they really are.

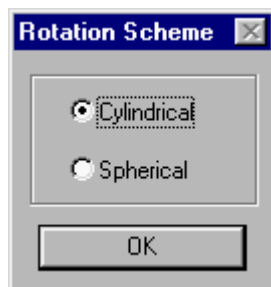
Most attributes for 3D scatter plots are similar to those for 2D scatter plots, as discussed above. However, three entries in the attributes list below are specific to 3D plots.

Figure 10.14
3D view attributes



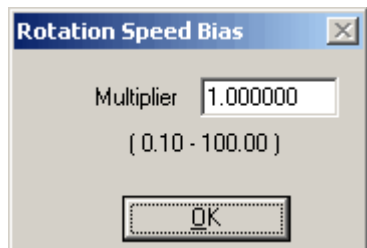
You can choose the 3D rotation scheme, as shown in the dialog box below. The cylindrical setting gives the impression that data points are contained in a cylinder or can, and the mouse action rotates the can in the direction of the cursor movement. Spherical rotation functions as if the data points were enclosed in a large ball, sitting on a flat surface, and the mouse action rolls the ball from the top.

Figure 10.15
Rotation Scheme dialog



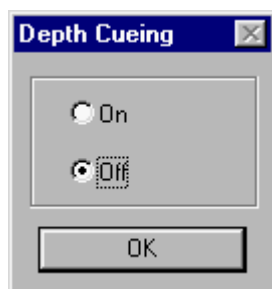
The speed of the computer processor has some effect on the speed of rotation in the 3D plot. You can modify the default speed accordingly via the Rotation Speed Bias, as shown in the following figure.

Figure 10.16
Rotation Speed Bias dialog



The remaining attribute conveys a sense of 3D on a two-dimensional computer screen. When depth cueing is turned on (see figure below), points lying behind the plane of the screen are "dimmed" to suggest being behind points in front. Note that depth cueing is not apparent for highlighted points because the plot symbol for a selected point overrides the depth status.

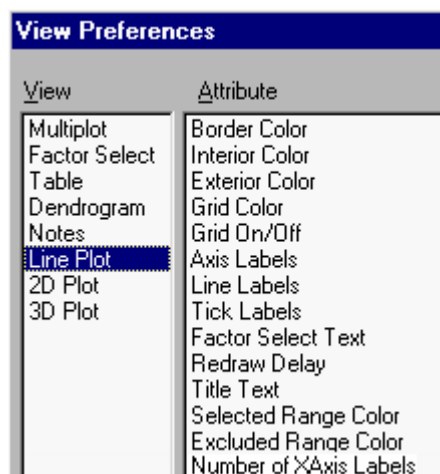
Figure 10.17
Depth Cueing dialog



Line Plot

The first group of attributes for line plots, shown in [Figure 10.18](#), have already been described. Refer to [Figure 10.3](#) and [Figure 10.5](#) for a discussion of setting color attributes and changing text attributes, respectively.

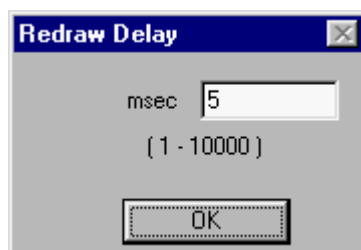
Figure 10.18
Line Plot attributes



The line plot has additional unique attributes. The colors of the range selection and excluded region selection are selected as previously described. Line plots have titles whose font and color can be chosen.

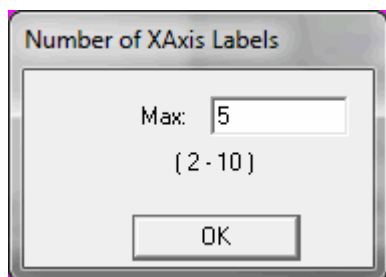
The Redraw Delay setting is the interval between the display of successive lines when triggered by the Redraw button, explained in [“Redrawing Traces” on page 12-19](#).

Figure 10.19
Redraw Delay dialog



The number of x-axis labels on a line plot can be controlled by setting a parameter as shown below.

Figure 10.20
Number of X-axis
labels dialog



Factor Select Plot

All of the attributes in the Factor Select plot preference suite are of types already described. Refer to [Figure 10.3](#) and [Figure 10.5](#) for a discussion of setting color attributes and changing text attributes, respectively.

Figure 10.21
Factor Selection
attributes

View Preferences	
View	Attribute
Multiplot	Border Color
Factor Select	Grid Color
Table	Interior Color
Dendrogram	Exterior Color
Notes	Grid On/Off
Line Plot	Axis Labels
2D Plot	Line Labels
3D Plot	Tick Labels

Dendrogram

The dendrogram, a special view in Pirouette, has several appearance settings. The colors of the dendrogram “tree” itself, as well as the reduced size overview are set via a dialog box of the type shown in [Figure 10.3](#), as are the background color, the color of the “leaf” or end points (shown for samples which are selected) and the color of the similarity line.

Figure 10.22
Dendrogram
attributes

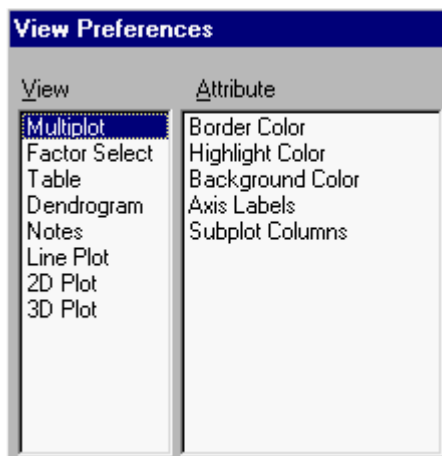
View Preferences	
View	Attribute
Multiplot	Border Color
Factor Select	Tree Color
Table	Overview Zoomed Color
Dendrogram	Background Color
Notes	End Point Color
Line Plot	Info Text
2D Plot	Leaf Labels
3D Plot	Tick Labels

Text in the dendrogram is set via a dialog box like that shown in [Figure 10.5](#). These attributes include the sample/variable “leaf” labels, the tick labels on the similarity axis and the informational text in the lower right corner of the dendrogram window.

Multiplot

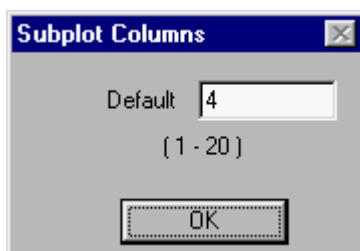
Multiplots are special arrays of 2D scatter plots. The 2D plot attributes are governed by the settings for that view. However, you have some control over a few additional features of the multiplot view, as shown below. The Highlight Color attribute controls the color of the highlighted subplot (*i.e.*, its border), and the Axis Labels attribute controls the font for the labels placed adjacent to each subplot. How these and the other color attributes are set has been previously described.

Figure 10.23
Multiplot attributes



Finally, you can specify the default number of subplots to be displayed for newly-created multiplots via the Subplot Columns dialog box.

Figure 10.24
Subplot Columns
dialog



Notes

The font of the text shown in the Notes windows that accompany all computed results are set by the standard means already described. A preview of appearance of these attributes is shown in the preferences window, as shown below.

Figure 10.25
Notes attribute

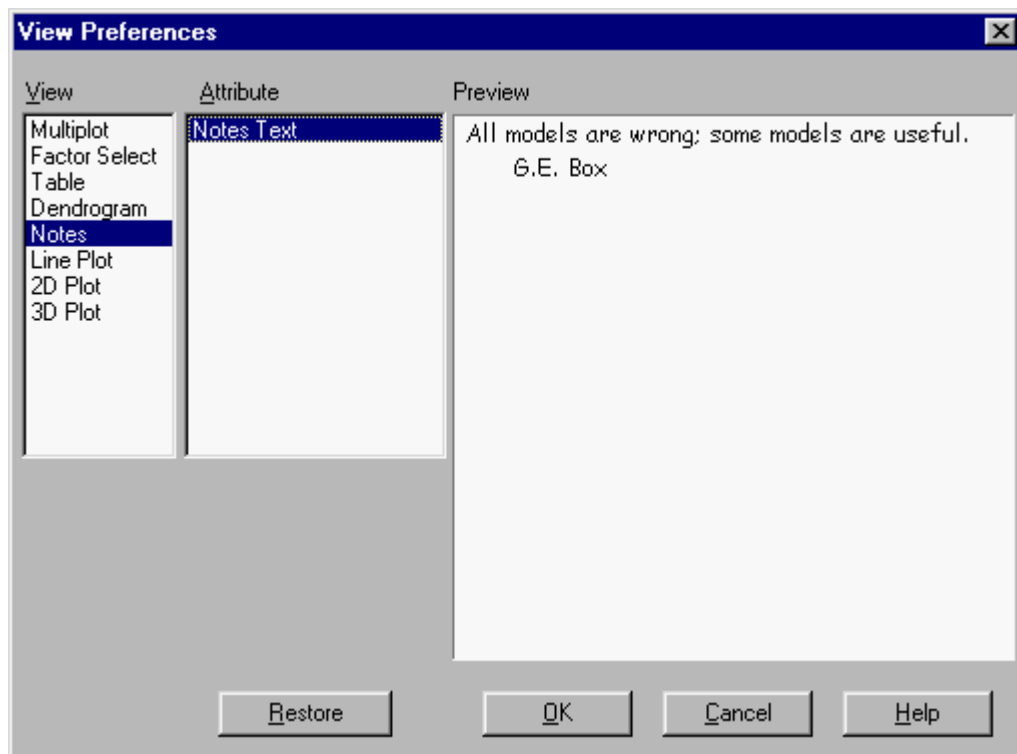
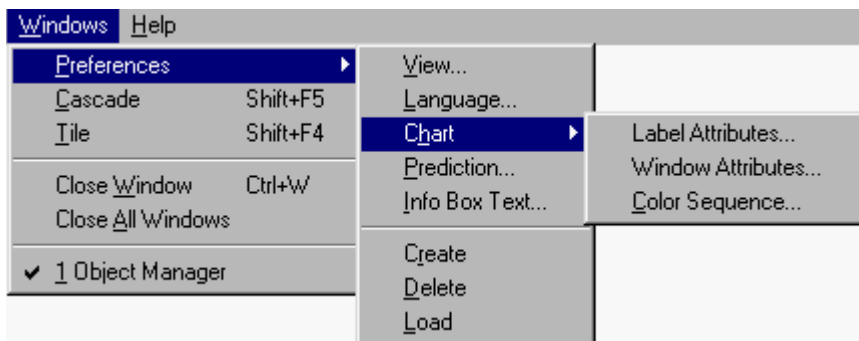


Chart Preferences

The preferences discussed above are specific to the type of plot view. A few additional preferences establish defaults for the plot windows themselves or for generic attributes of plot features. This group of preferences is accessed by the Windows > Preferences > Chart menu item:

Figure 10.26
Chart Preferences submenu



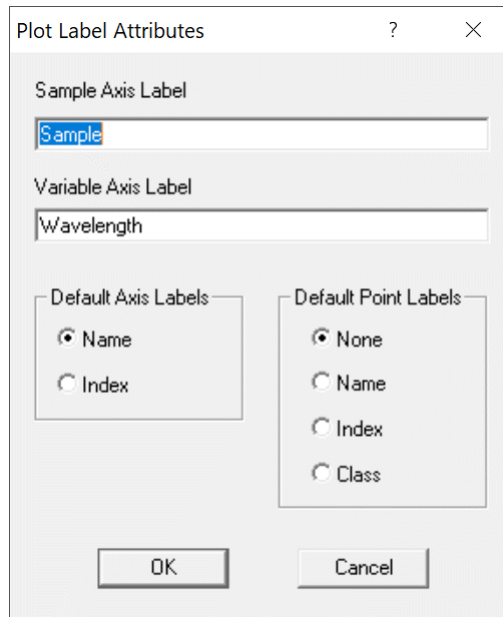
LABEL ATTRIBUTES

From the Label Attributes dialog box, you have the option of setting several defaults which will affect all plots that are created, automatically or manually through “drag and drop”.

Axis Label Modes

You can specify that data points in new scatter plots be shown with or without labels, in one of the three modes offered by the following dialog box. In addition, you can ask Pirouette to initially display axis labels with their names or by their index numbers.

Figure 10.27
Plot Label Attributes dialog



Axis Labels

Using the label fields in this dialog box, the spectroscopy user, for example, can display an appropriate label (e.g., “wavelength”) for spectral plots. Similarly, use this dialog box to specify a name for the sample axis label.

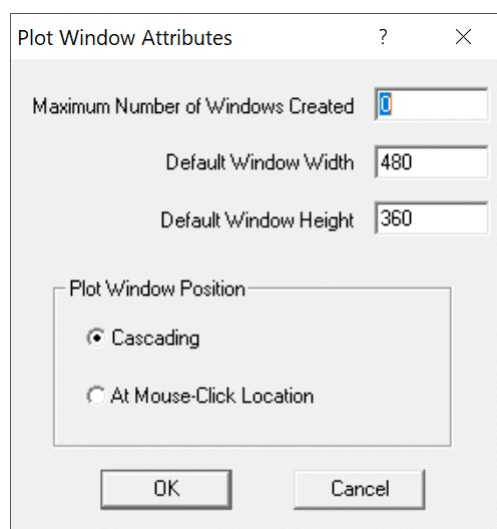
WINDOW ATTRIBUTES

Pirouette allows you to control several aspects of how results windows are displayed through the Window Attributes dialog box.

Number of Plot Windows

By default, Pirouette displays no computed results for an algorithm when it finishes. This avoids spending a considerable time constructing array plots which you may often immediately close. The Maximum Number of Windows Created value shown in the figure below allows you to dictate the number of results windows shown following completion of a batch run. For example, if you set this value to ‘2’ and configure 3 different PLS runs, only results of the first two are shown immediately. Of course, you can always drag from the Object Manager for any result not initially displayed.

Figure 10.28
Plot Window
Attributes dialog



Size of Plot Windows

When creating a new plot window, Pirouette by default prepares a window of size 640 by 480 pixels. You can override the default size for new windows by using the above dialog box. If you leave either of these values at 0, the size reverts to the original Pirouette default.

Location of Plot Windows

When you drag a new plot window from the Object Manager, its initial position is governed by a standard cascading mechanism: it is slightly offset horizontally and vertically from the last created window. From this dialog, you can choose to instead position the upper left corner of windows at the cursor ‘drop’ position.

COLOR SEQUENCE

Color sequencing affects Pirouette’s various graphic displays, most obviously the color bar associated with similarity settings in the dendrogram. In addition, points in 2D and 3D plots and traces in line plots are assigned colors derived from the Color Sequence. See [“Creating Class Variables” on page 12-27](#) for a discussion of how colors are assigned in the dendrogram and [“Plot Colors” on page 12-34](#) for an explanation of how colors are assigned in scatter and line plots. The color sequence is set via the dialog shown below.

Figure 10.29
Color Sequence
dialog



Note that there are two colors lists: Current and New. Any changes appear on the New list. You can add a new color or delete an existing color from the list by clicking on a button. When you click on Add, the color palette is presented as in [Figure 10.3](#). In addition, you can modify an existing color by double-clicking on the color in the New list. Although the total number of colors in a sequence is unlimited, you should consider specifying fewer than a dozen colors, depending on your video card's color depth.

Note: Colors are mapped directly to values in the active class. If values are continuous (e.g., 1, 2, 3, 4, 5), point colors follow the sequence exactly. If values are discontinuous (e.g., 5, 10, 16, 28), two classes may have the same color—if, for example, the difference between two class values is greater by 1 than the number of colors in the sequence.

Other Preferences

PREDICTION

Unlike during training, options that are used by algorithms when performing predictions are set in a dialog box separate from the Predict Configure. These options are not stored in the corresponding model but are applied whenever a prediction is made using the associated model. For example, you can override the probability setting in a model and make predictions based on a more lenient or more restrict qualifier. To access the dialog box shown below, go to Windows > Preferences > Prediction.

Figure 10.30
Prediction
parameters dialog
box

Prediction Parameters

Classification

Probability: 0.950000 (0 - 1)

Augment Sample Residual

Mask Variable: Mask

Calibration Transfer Type: None

Window Size: 0

Use class mean

Regression

Probability: 0.950000 (0 - 1)

Mask Variable: Mask

Calibration Transfer Type: None

Window Size: 0

OK Cancel

These settings are grouped by whether they apply to classification or to regression algorithms. Refer to the respective chapters for greater detail of the options and how they are used.

Some of the classification set of options also refer to PCA. For example, PCA and SIM-CA, which are closely related algorithms, contain two parameters which control the size of the scores hyperbox and affect membership/outlier decisions for prediction samples; see [“Augmenting the Sample Residual in Prediction” on page 5-30](#) and [“Class Probabilities” on page 6-27](#).

Finally, although PLS-DA is a classification algorithm, it is closely related to the PLS regression algorithm, and the regression settings above apply.

INFO BOX FONT

This setting controls the attributes of text appearing when the right mouse button is clicked on any entity in the Object Manager and when Model Info is displayed in the Configure Prediction and Save Models dialog boxes. The font attributes for these text displays are set in a dialog box like that shown in [Figure 10.5, on page 10-9](#).

Note, however, that text attributes of the Notes object accompanying each set of computed results are not controlled by the Info Box Font, rather by its own view preference; see [“Notes” on page 10-15](#).

STICKY FEATURES AND DEFAULT SETTINGS

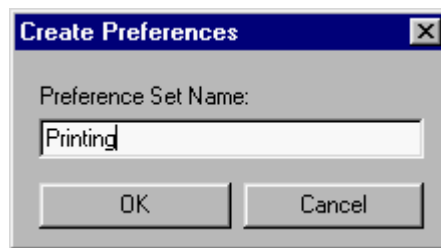
Pirouette is shipped with an initial set of preferences. To revert to the Infometrix settings after making changes, click the Restore Default button present in all preferences dialogs. Many parameter settings not set via a preference dialog are instead “sticky”, that is, the next time you use the feature, its previous setting is preserved. For example, when you

load a data file and set the File Type to be ASCII, the next time you access the Open Data dialog box, the file type will still be set to ASCII.

Preference Sets

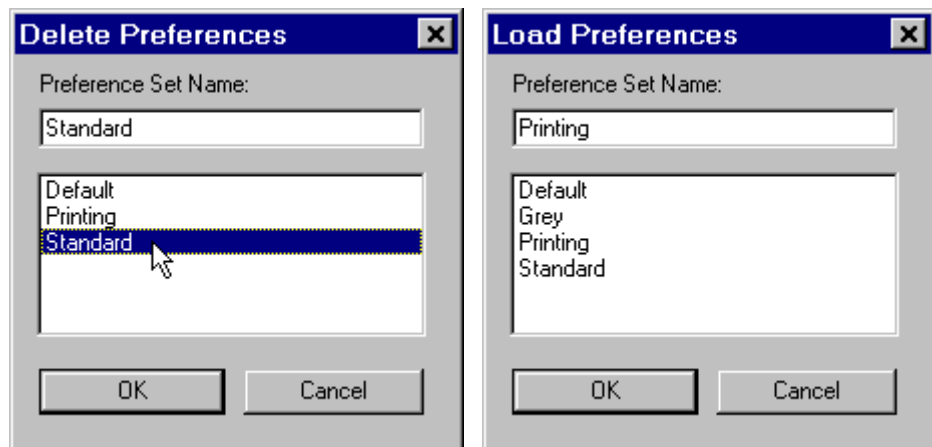
The last three items on the Windows > Preferences menu manage preference sets customized for special uses, such as printing. When you find settings you would like to preserve, choose Create and assign a name as shown in the following figure

Figure 10.31
Create Preferences dialog



Subsequent preference changes accumulate under the assigned name. Because any changes made to preferences affect only the currently active set, choose the Create menu item to start a new set, then proceed to change preferences. That way, the set active before you created the new set is not perturbed. You may delete or switch to a different preference set as shown below.

Figure 10.32
Deleting and loading preference sets



Language

Menus, text in dialogs and messages and warnings have been localized in Pirouette. Selection of the language to use is done in a program external to Pirouette (prior to version 4.0, this was done via a dialog in the Pirouette preferences). To choose the language in which you wish to display these parts of the Pirouette interface, select Start > Programs > Infometrix > Language Switcher.

10 The Pirouette Interface: Language

Figure 10.33
Language Selection
dialog box



The languages supported include the following:

- English
- French
- German
- Italian
- Japanese
- Portuguese
- Spanish

After you change the language, you must quit, then restart, Pirouette for all of the appropriate text to appear in the interface.

Note: Limited users cannot change the language setting. Ask your IT person or other user with administrator privileges to change the language for you.

Object Management

Contents

The Object Manager Window	11-1
Charts	11-7
Subsets	11-9






In Pirouette, the term *object* refers to either the raw data for any subset in the open data file or an algorithm result. Because you will often create and run algorithms on more than one subset of your data and because most algorithms produce multiple results, it is easy to generate a large number of objects for each file. The Object Manager keeps them organized and provides an easy to use, visual record of your work within Pirouette. For example,




- It lists the existing subsets and the algorithms which have been run on them
- It shows the current view of any existing charts
- It provides a means to assemble disparate results in a single window

The Object Manager Window

The Object Manager window is composed of an Objects tree that organizes all computed results for every subset contained in a Pirouette file. Each tree has branches nested to show successively more specific information. The representation of items within the tree are iconic. The following table lists all Object Manager icons and describes their correspondence to objects and plot objects.

Table 11.1
Object Manager
icons

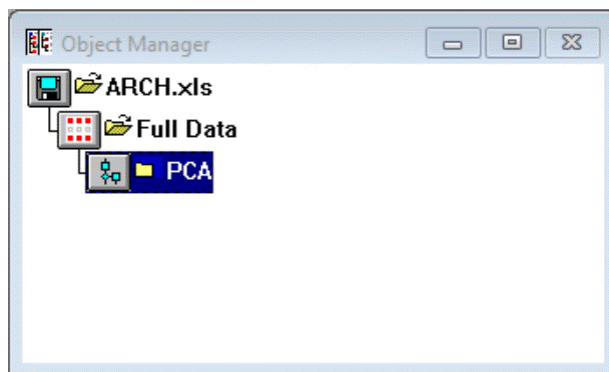
Icon	Description
	<i>Object Tree icons</i>
	File root
	Data subset
	All results from computation with a given algorithm
	Group (or subset) of results from a given algorithm
	Algorithm result which is a matrix

Icon	Description
	Algorithm result which is a vector
	Dendrogram result
	Notes

NAVIGATION

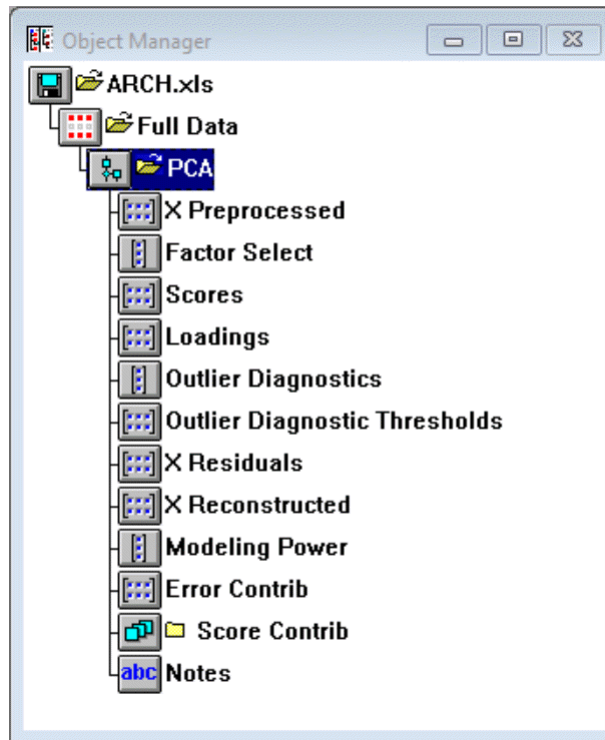
The Object Manager window can be minimized but not be closed. After an algorithm has been run, the file root icon shows an attached folder icon, initially in the closed aspect.

Figure 11.1
Object Manager after
running an algorithm



Moving up and down the object tree is much like navigation in the Windows File Manager or Explorer. To open a folder, double-click on its icon. To close a folder, double-click on the open folder icon.

Figure 11.2
Object Manager with
open algorithm
folder



To open all folders, click on the right arrow ribbon button; it opens the Object Manager tree one level. Similarly, each click on the left arrow closes the trees one level.

NAMING CONVENTIONS

Each level in the Object Manager hierarchy has a naming convention.

Set Level

Sets appear at the level just below the root icon. In [Figure 11.2](#) only one set, called *Full Data*, exists. Subsets formed either by dragging the file root icon to the work area or by create exclude/include operations are assigned default names which you can change; see [Renaming Objects on page 11-6](#).

Algorithm Level

Algorithm folders are nested below the set on which the algorithm was run. Repeat runs on a subset are distinguished by an appended sequence code. Algorithm folders, like sets, can be renamed.

Note: Note that you cannot rename a subset or an algorithm folder with an already existing name.

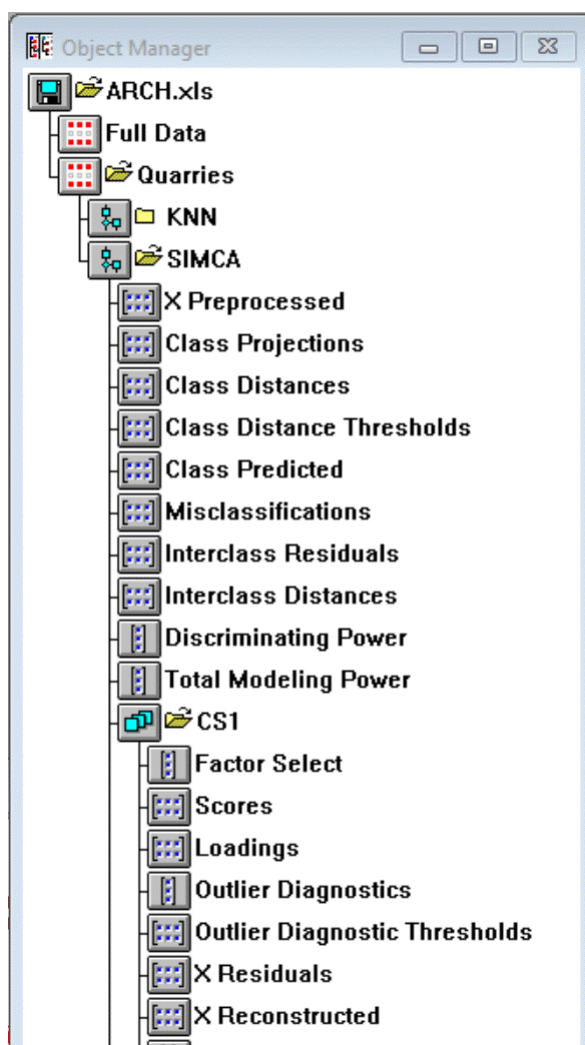
Results Level

Each algorithm execution produces one or more objects which are stored in an Algorithm folder. The Object Manager, when expanded to show the third level, lists these objects by name. The names usually correspond to the matrix entity produced by the computa-

tion, for example, PCA Scores. However, some results are collections of vector entities which have important relationships to each other, for example, PCA Outlier Diagnostics, which includes Sample Residual and Mahalanobis Distance vectors. Object Manager entries at these levels cannot be renamed.

Several algorithms produce results tied to a Y variable or a class category. In such a situation, a folder is shown at the third level, indicating that the results are grouped and individual results appear as level four items. Thus, a SIMCA Scores object is listed underneath the category's value, indicating that the scores are those from only that category. In the following figure, there is a display of SIMCA results listed under CS1, which is the 1st class category in the data set.

Figure 11.3
Object Manager with
SIMCA results



Object Information

Names may not always be specific enough to relate the history of the object. Therefore, all algorithm results are accompanied by another object known as the Notes. This object contains, by default, a description of the computed algorithm results in a separate text window; you can also insert additional information about your analysis in this window.

If you click on an Object Manager icon with the right mouse button, a small window opens containing auxiliary information. The type of information depends on the object selected.

- Click on the root icon to see the full path to the data file
- Click on the set icon to see the dimensionality of the set
- Click on the algorithm folder icon to see run configuration details
- Click on the group icon (Y variable or class category name) to display the “path” to the group, *i.e.*, the set and algorithm names
- Click on the result icon to display the “path” to the result

An example of the algorithm information is given below.

Figure 11.4
Object information

```

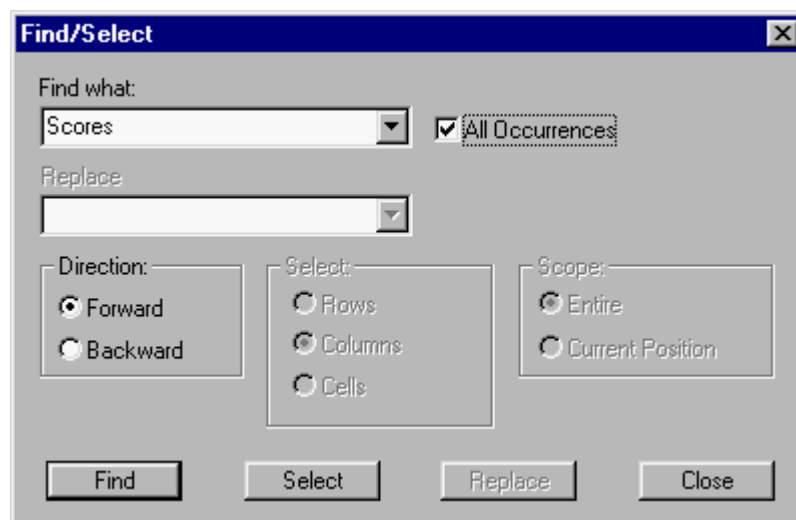
                                SIMCA
# of Included Samples:           63
# of Included X vars:           10
Class Variable:                 Quarry
Preprocessing:                 Autoscale
Scope:                         Local
Maximum factors:               3
Optimal factors:               2, 3, 3, 3
Prob. threshold:               0.9500
Calib Transfer:               Not enabled
Transforms:
  None
Algorithm date:                01/04/2023 11:16:38.
User ID:                       n/a
ID Source:                     n/a
Windows Login:
```

A right mouse mechanism also exists for identifying plots in an array; see [page 12-3](#) for details.

FINDING OBJECTS

If the list of objects is short, it is easy to find the subset or algorithm results. However, after you have run many algorithms on a plethora of subsets, scrolling the Object Manager window becomes tedious. The Find dialog box facilitates searches.

Figure 11.5
Find dialog box



The basic options are Find and Select. The former moves the highlight forward or backward to the next instance of the text string. The Select button in combination with the All Occurrences check box is useful when you want to highlight all objects containing a given text string in the Object Manager display. This feature enables you to create a custom array plot of, for example, all scores from multiple runs of PCA using different preprocessing settings. Note that the “Find what” string is case sensitive.

Note: *The object(s) you wish to find must be visible in the Objects tree. Use the right navigation arrow in the ribbon to expand the tree to show as many levels of objects needed.*

If the result of your find operation highlights more than one result in the Object Manager, click down on one of the highlighted objects, then drag to the workspace and a custom plot array will be created with each highlighted object as a subplot in the array (see also “Custom Charts” on page 11-8).

If you want these multiple objects to each appear in its own window, hold down the Shift key just before releasing the mouse.

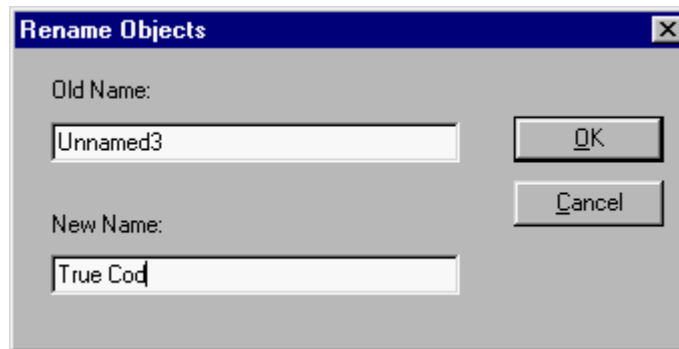
RENAMING OBJECTS

Pirouette automatically assigns a unique name to each new subset based on the root word *Unnamed*. This permits you to create a series of subsets without taking the time to rename them. However, eventually you will want to assign more descriptive names. To rename a subset,

- Click on the Set icon in the Object Manager
- Choose Rename from the Objects menu

and the dialog box shown below will be displayed.

Figure 11.6
Rename dialog box



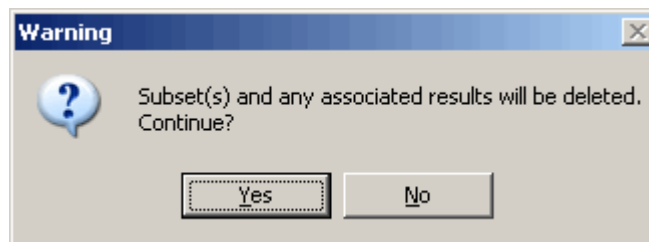
- Type in a new name for the set
- Click on OK

The revised name will appear beside the Set icon and in the title bar of corresponding plots. The set name also appears in other dialog boxes: the Predict Configure, Save Model and Save Objects dialogs. The same general procedure can be used to rename algorithm folders.

DELETING OBJECTS

When you save a file, Pirouette stores everything present shown in the Object Manager. To prevent algorithm results or subsets from being stored, click once on the object's icon, then select Delete from the Edit menu or press the Del key. Before Pirouette deletes the object, you are reminded, via the following dialog box, that this action is irreversible.

Figure 11.7
Warning dialog when deleting objects



When sets or algorithm folders are deleted, any existing plots containing those results are also deleted.

Note: *Retaining all sets and their computed results creates a sort of processing audit trail. Deleting objects, on the other hand, results in smaller data files.*

Charts

After an algorithm runs and you drag results to the work area, Pirouette displays the results in a Chart window. If you drag the algorithm folder, rather than an individual object, the chart will be a collection of subplots called an *array plot*. You can retrieve any chart from the Object Manager or create a custom array by drag-and-drop tactics. Clicking on a chart's go away box causes the chart to be deleted but you can always recreate it as described below.

CREATING CHARTS

To create a chart via a procedure commonly referred to as *Drag and Drop*,

- Click on the icon in the Object Manager
- With the mouse button held down, move the cursor out of the Object Manager window
- When the cursor is over a clear space in the Pirouette window (called the Pirouette workspace), release the mouse button



During dragging, while the cursor is over the Object Manager window, it resembles a hand carrying a chart:

When the cursor is over an allowed drop area, it resembles a hand releasing a chart:



Allowed drag and drop areas are unoccupied space in the Pirouette window, any existing chart and the ribbon area. For more information on charts and plots, see [Chapter 12, Charts](#), and read the next section, which discusses custom charts. Two possibilities for creating charts are:

- Drag and drop an Algorithm icon (a folder) to make a chart with all algorithm results. If you have put away (closed) the original chart created when the algorithm was run, this is how you would recreate the same view of the results.
- Drag and drop a single result icon to make a chart of only that result. In many situations, a chart with only a single result plot is more useful than having that view shared with other subplots and is quicker to generate.

CUSTOM CHARTS

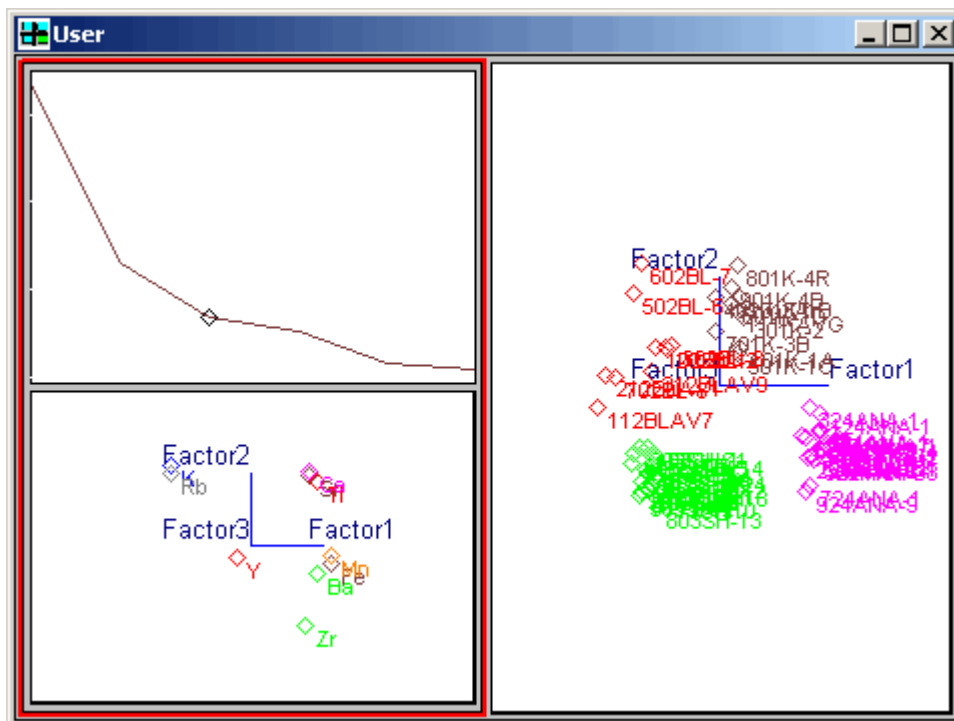
A powerful aspect of drag and drop chart creation is the ability to make custom charts. For example,

- Drag and drop an object icon onto an existing chart to add that new object to the collection of plots already in the chart.
- Highlight more than one object or chart icon. Drag and drop the group to create a new chart or add it to an existing chart.

Object Manager entities below the file root icon can be dropped into an existing chart. Thus, custom charts can contain a combination of subplots. In fact, you can add objects to zoomed subplots so that when the chart is fully shrunk to its array view, some subplots are given more importance than others. For example, [Figure 11.8](#), which emphasizes the scores plot, was created via the following steps:

- Click on the Factor Select object in the Object Manager
- Ctrl-click on the Scores object in the Object Manager
- Drag the two icons to the Pirouette desktop to create a new “User” chart with these two objects side-by-side
- Zoom the Factor Select subplot by pressing Enter or double-clicking the subplot
- Click on the Loadings object in Object Manager
- Drag the Loading object onto the zoomed chart
- Unzoom the Factor Select plot by pressing Ctrl-Enter or Shift-double-click

Figure 11.8
Custom chart with an
array of an array



Subsets

Subset creation is a powerful means to investigate multivariate data. We may wonder if a result is dominated by anomalous samples or variables. By excluding suspect samples and/or variables and rerunning algorithms, we can determine the general applicability of a result. Pirouette allows exclusion subset creation from either tabular (see “[Creating Subsets from Tables](#)” in Chapter 13) or graphical (see “[Creating Subsets from a Graphic](#)” in Chapter 12) views. Once a subset has been created, it appears in the Object Manager with a default name of *UnnamedN*, where N is a positive integer.

Note: *When you save a data set in the PIR format, all subsets are also saved. The next time you open that file, these subsets will be listed in the Object Manager.*

It is also possible to create a new subset from the Object Manager. Drag and drop the disk icon onto the work area to display a new window containing a table view of the data set with all samples and variables included and produce a new Set icon in the Object Manager tree. This is a mechanism to create a subset in which all rows and columns are included.

Finally, computational methods can be used to create subsets. These are discussed next.

SAMPLE SELECTION

In some applications, data are easily generated and it is possible to end up with a data set containing so many samples that processing may bog down. Or perhaps the data space variation can be represented by just a small subset. Or perhaps you have unbalanced numbers of samples in several categories. Each of these scenarios can benefit from sample selection.

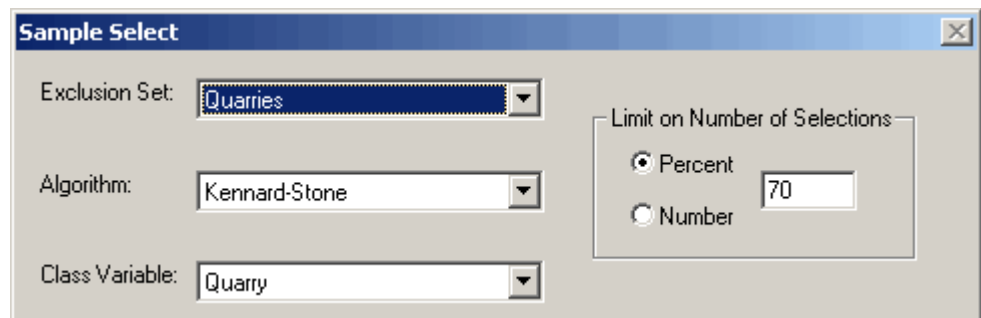
- Choose Process > Select Samples

to present the dialog box shown in Figure 11.9. Once you have specified the appropriate parameters, a new subset is created that contains only the samples found by the selection algorithm. Subsets created in this manner are named using the following pattern:

SetName [ALG-NN%-Class]

where SetName is derived from the name of the exclusion set that is the basis for the selections, ALG is a mnemonic derived from the method name, NN is the number of samples requested (% is added if that option is active), and Class is the name of the class variable used for grouping samples, if any.

Figure 11.9
The Sample Selection dialog



- Choose Exclusion Set from the drop-down list

Samples can be selected from any available subset. Only the subset's included variables and samples are considered by the selection method.

- Choose the category to use from the Class Variable drop-down list

If class variables exist, they are listed here; they drive category by category sample selection. When the class variable is not set to None, the selection algorithm will be repeated for each category.

- Choose how many samples to retain in the new subset

You can control how many samples to select based on percentage or absolute number. If you chose to make selections on a per-category basis, then the percentage and number choices are applied to the samples in each category, independently. If you choose a number larger than the number of samples in a category, all samples from that category will be retained.

The sample selection algorithms are described below.

Kennard-Stone¹

This method finds samples that are most dispersed across the data space.

Euclidean distances are calculated among samples. The two samples with the largest intersample distance are selected first. Then, additional samples are added to the list by two

criteria: for each sample not yet in the list, determine its nearest neighbor among the current selections; select that sample whose nearest neighbor is of the largest distance.

Orthogonal Leverage²

This method finds samples of greatest influence within the data space.

Leverages are computed among all samples, and the sample of greatest leverage is chosen. The remaining samples are orthogonalized against the selected sample, then the process is repeated until the desired number of samples are selected.

PCA Hypergrid³

This method finds samples that are the most uniformly distributed in a reduced factor space.

A PCA is run on the sample set, and the factors are trimmed, reducing its dimensionality. In the trimmed scores space, a hypergrid is formed by dividing each factor dimension proportionally. Samples are selected by choosing one sample nearest the center of each block formed from the hypergrid.

Random

As implied by its name, this method will create a subset of the desired number of samples by a random process. Use random subsets to test reliability of multivariate models. Because it is likely you will want to challenge the model with another random set of samples not present in the model set, another control is offered in the dialog—Create complement set—that creates an additional subset from the remaining samples in the target exclusion subset but with all variables included. Thus, the complement set is ready for prediction.

VARIABLE SELECTION

Factor based methods, when applied to data of high signal to noise ratio, are very good at isolating signal from noise, resulting in dimensionality reduction. However, if very many correlated variables are present, this ability can be confounded. Thus, for some data it can be beneficial to remove unimportant variables. Pirouette offers some simple variable selection routines.

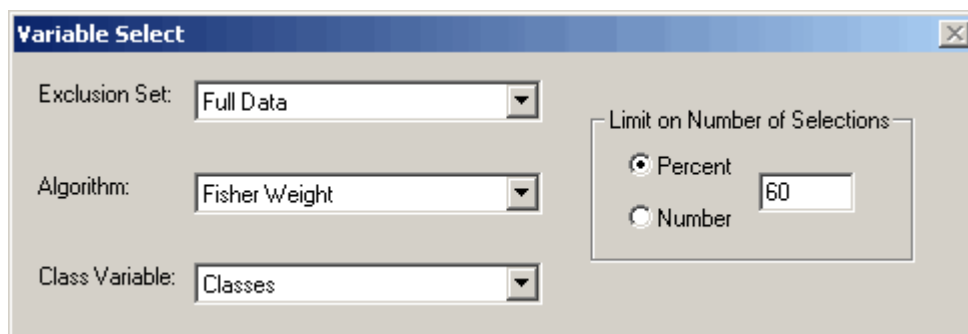
- Choose Process > Select Variables

to present the dialog of [Figure 11.10](#). Once you have specified the appropriate parameters, a new subset is created that contains only the variables found by the selection algorithm. Subsets created in this manner are named using the following pattern:

SetName [ALG-NN%-Class]

where SetName is derived from the name of the exclusion set that is the basis for the selections, ALG is a mnemonic derived from the method name, NN is the number of variables requested (% is added if percent is used), and Class is the name of the class variable used by the algorithm, if relevant.

Figure 11.10
Variable selection
dialog



- Choose Exclusion Set from the drop-down list

Variable can be selected from any available subset. Only the subset's included variables and samples are considered by the selection method.

- Choose how many variables to retain in the new subset

You can control how many samples to select based on percentage or absolute number.

The variable selection algorithms are described below.

Standard Deviation Rank

The standard deviation is computed for each variable, then they are sorted in decreasing order of standard deviation. The first N variables are retained as included in the new subset, where N is the number of variables selected (see above).

Fisher Weight and Variance Weight⁴

These methods measure the importance of a variable in discriminating among two or more categories. Fisher weight ratios the intercategory means to the intracategory variances while the Variance weight replaces the means with the intercategory variances. To use either of these methods,

- Select a Class variable

before running the algorithm.

References

1. Kennard, R. W. and Stone, L. A. "Computer aided design of experiments." *Technometrics*. 1969; 11(1):137-148.
2. Wang, Yongdong; Veltkamp, David J., and Kowalski, Bruce R. "Multivariate instrument standardization." *Analytical Chemistry*. 1991; 63:2750-2756.
3. Carpenter, S. E. and Small, G. W. "Selection of optimum training sets for use in pattern recognition analysis of chemical data." *Analytica Chimica Acta*. 1991; 249:305-321.
4. Sharaf, M.A.; Illman, D.L.; and Kowalski, B.R.; *Chemometrics* (Wiley: New York, 1986), p. 195.

Charts

Contents

Creating Charts	12-1
Window Titles	12-3
Pirouette Graph Types	12-4
Scatter Plots	12-5
Line Plots	12-13
Multiplots	12-20
The Dendrogram	12-22
Linking Views	12-28
Creating Subsets from a Graphic	12-32
Plot Colors	12-34

In this chapter we show how interactive graphics enable you to better understand complex data. Multivariate approaches are the key to turning such data into information, and visualization is a means to discovering patterns in your data. This chapter describes Pirouette's graph types, explaining how to manipulate each. Included are discussions of how to link results in different windows and how to create data subsets from a graphic.

Creating Charts

Algorithms typically produce more than one result. Thus, chart windows produced by an algorithm often contain an array of subplots, each representing an object. Pirouette's standard charts are probably adequate for most data analysis tasks. However, many situations demand the flexibility to create data and result views that more perfectly tell a story. Therefore, Pirouette lets you create custom charts either from the Object Manager or via the Drop button.

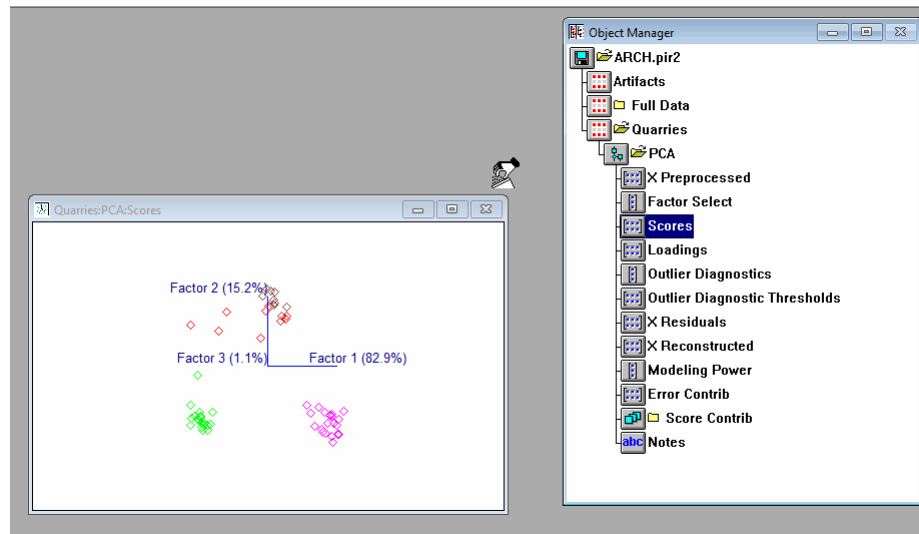
CREATING CHARTS FROM THE OBJECT MANAGER

Clicking on an icon in the Object Manager highlights it, showing its selected state. If you then click and drag, the cursor changes to the Drag form (described in [Table 10.8, "Pirouette cursors," on page 10-6](#)), indicating that the contents have been picked up and can be transported to a destination. Moving the mouse out of the Object Manager window changes the cursor to a Drop form. Releasing the mouse button drops the object and cre-

12 Charts: Creating Charts

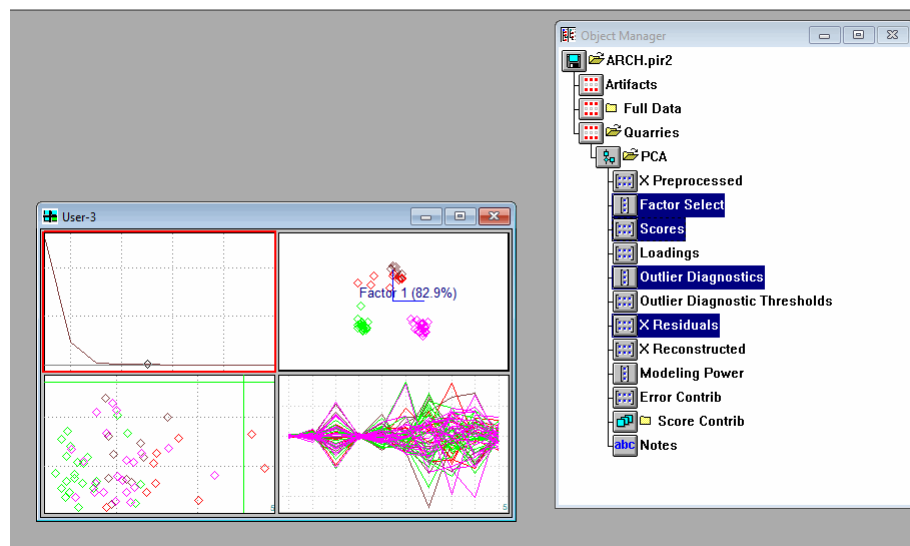
ates a new chart. This process is commonly called *drag and drop*. A new chart can be created by dragging and dropping from the Object Manager. The figure below demonstrates a drag and drop.

Figure 12.1
Creating a chart from
the Object Manager



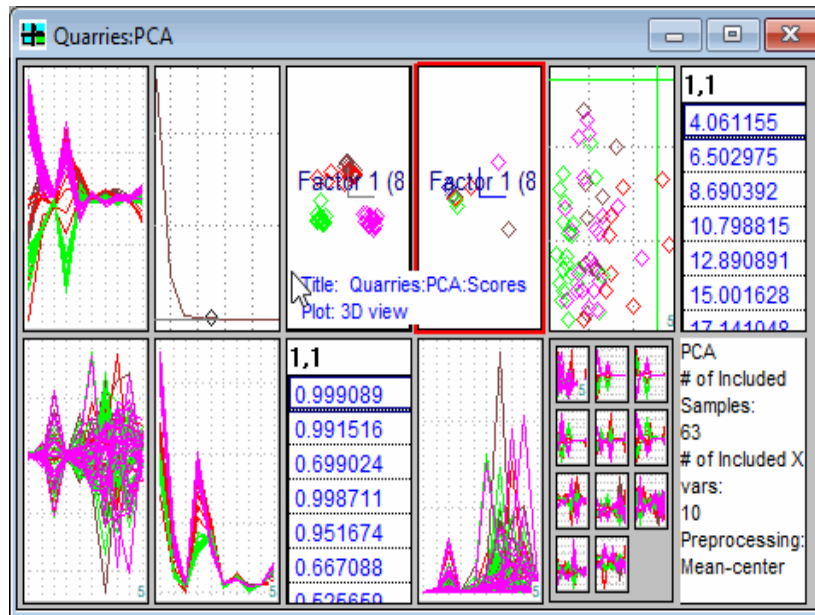
If you hold down the Ctrl key while clicking on an Object Manager entry, any previously highlighted icon remains highlighted and additional selections accumulate. If more than one object is highlighted before the drag and the Shift key is pressed right before dropping, one chart is created for each object. If the Shift key is not pressed upon dropping, the resulting chart contains subplots of every object selected. Such a collection of plots is called an array plot. This process is illustrated in the figure below.

Figure 12.2
Creating an array
plot by dragging
multiple objects



It is sometimes difficult to identify subplots in an array. Clicking with the right mouse button on a subplot will present an information box listing certain key properties of the object. The following figure is an example of this properties display.

Figure 12.3
Subplot information



Before you can interact with a subplot, it must be zoomed; see [Table 12.4, “Zooming and Unzooming,”](#) on page 12-20 to learn how to accomplish this.

CREATING CHARTS WITH THE DROP BUTTON

Pirouette provides another mechanism for creating custom charts: accessing the Grabber tool via the Drop button. To use this method, you must have the chart visible (not minimized). This is a convenient way of precisely duplicating a graphical view, yielding two copies with the same zoom level and orientation.

- Click on the source chart window title, making it active
- Click on the Drop button in the ribbon
- Move the cursor over the source chart
- Click-drag the cursor to the Pirouette desktop
- Release the mouse button.

When you grab a chart, whatever is displayed is dragged. Thus, if a chart array is the current window, then the entire array of subplots is copied. If, however, you have zoomed one subplot, only that subplot is copied to the Pirouette work area.

Window Titles

Window titles have the general form:

NAME:ALGORITHM:VARIABLE:OBJECT;SUBOBJECT;#

where the pieces are defined in the following table.

Table 12.1
Window title
components

Component	Description
NAME	Subset name
ALGORITHM	Algorithm abbreviation
VARIABLE	Descriptor for the regression (Yn/name) or category (CSnumber) variable.
OBJECT	Result's abbreviation
SUB-OBJECT	Sub-result's name or abbreviation
#	Copy counter

The first two components are self-evident and can be also be gleaned from the corresponding Object Manager entry. The third item, VARIABLE, appears only in algorithms which produce results tied to Y or C variables. For example, PCR and PLS produce some results for each dependent (or Y) variable. If more than one Y variable is included, the naming convention distinguishes the Ys. Similarly, SIMCA generates results tied to each category in the active class variable. The VARIABLE part of the name has two pieces: the CS prefix, telling that this a class specific result. The second part is the actual category value. PLS-DA combines the variable descriptor to include both the Y index and the CS value in the class variable used.

The OBJECT piece of the window title is often an abbreviated form of the result name. Some algorithms may have sub-results, such as the score contributions, which reside in folders inside the main result object. The final item is an integer denoting the i^{th} copy of a window. Thus, if you drag and drop a subset twice, the second window title is SUB-SETNAME;2.

The naming convention in [Table 12.1](#) applies only to charts created either automatically or by dragging a single subset or algorithm result icon to the Pirouette work area. Chart arrays created by dragging and dropping a variety of results and/or existing charts have the generic title USER.

Arranging Windows

You can minimize and rearrange chart windows. The Windows menu contains the Cascade and Tile items found in most programs. By default new charts cascade relative to the last-created chart. However, you can specify that a new chart window appears at the drop position; see [“Plot Window Attributes dialog” on page 10-18](#).

Pirouette Graph Types

Pirouette objects can be viewed in one or more forms:

- 3D plot
- 2D plot
- Line Plot, including factor selection line plot
- Multiplot
- Array Plot
- Table

- Dendrogram

Default views for algorithm results have been defined within Pirouette. However, view types can be changed by clicking on the appropriate ribbon button. Raw data (including subsets) can be plotted in all views except the dendrogram. The dendrogram, specific to the HCA algorithm, cannot be converted to any other view and non-dendrogram objects cannot be converted to its view.


The availability of a view type depends on the dimensionality of the object's table view. Each row/column in a table has an index which is its row/column number. In the next section, the discussion of view types and how to interact with each is often couched in terms of row/column indices.

Scatter Plots

Scatter plots contain a column plotted against another column or a row index; every row in the table view of the object then becomes a single point in the plot. Often scatter plots are called 2D or 3D plots. A 2 dimensional scatter plot is initially composed of the first column on the X-axis and the second column on the Y-axis. A 3 dimensional scatter plot initially has the third column on the Z-axis.

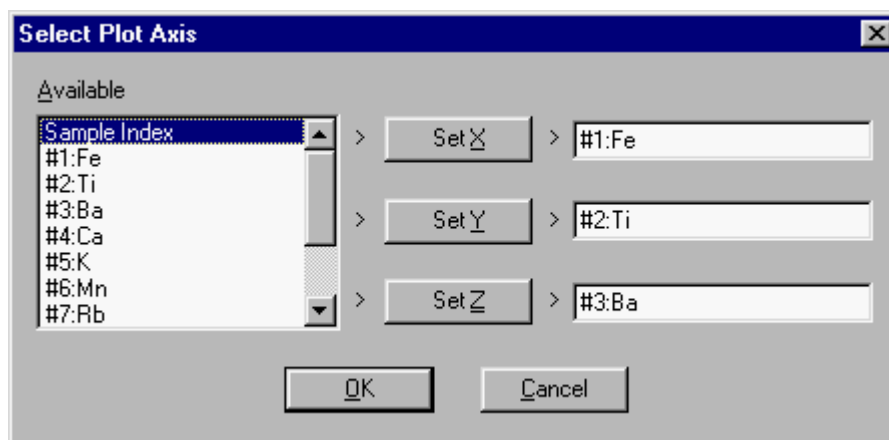
SPECIFYING AXES

To change the columns displayed in a scatter plot,

- Click on the Selector button  in the Ribbon

and the following selection dialog box will be displayed.

Figure 12.4
Scatter Plot Selector
dialog




Highlight an entry in the Available list and click the appropriate Set axis button. When you click OK, the plot will be updated to show points as a function of the chosen variable. If you are working with a 2D plot, the Set Z button will not appear.

SELECTING POINTS

Many of the interactive operations available in Pirouette require selection of points in scatter plots. To select points in a 2D or 3D scatter plot,

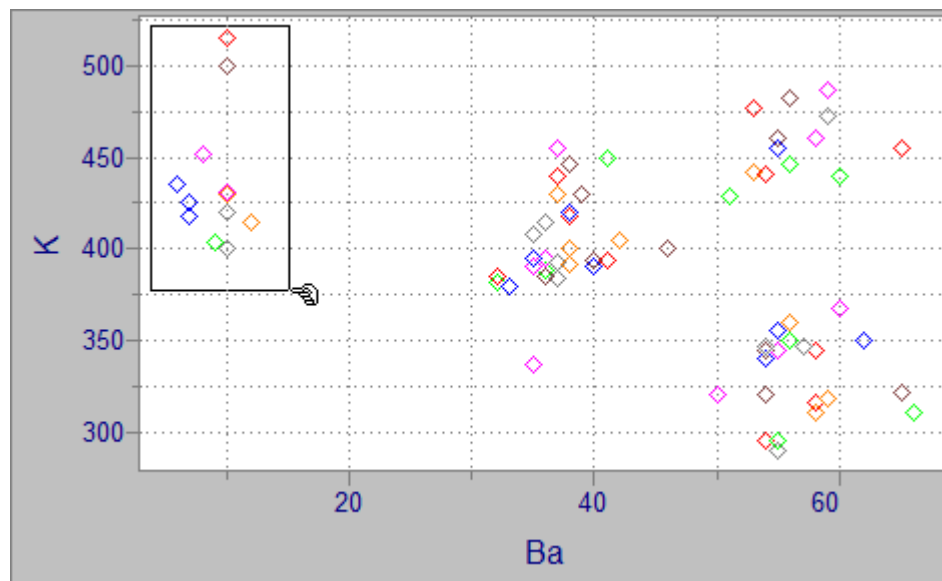
12 Charts: Scatter Plots

- Click on the Pointer button  in the Ribbon

Move the cursor over the plot until the Pointer fingertip is just above and to the left of the first point you want to include in the selection, then hold the mouse down and drag down and to the right until all the points you want to select are contained in the rubber box.

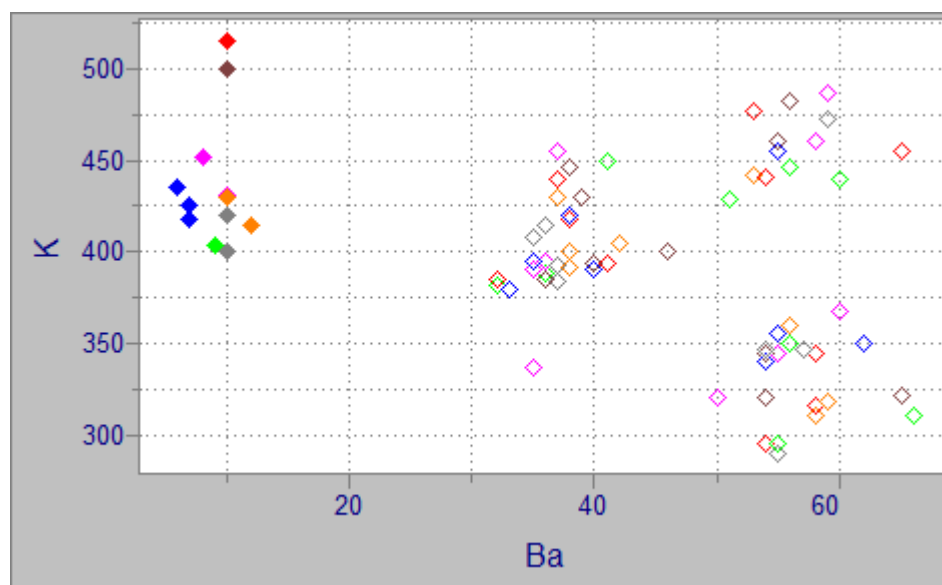
Figure 12.5 shows how to select the points in the left side of the graphic.

Figure 12.5
Selecting points with
the Pointer tool



As shown in the next figure, all points within the rubber box are highlighted, *i.e.*, the filled diamond symbol indicates the selected status of the sample points on the left side of the graphic.

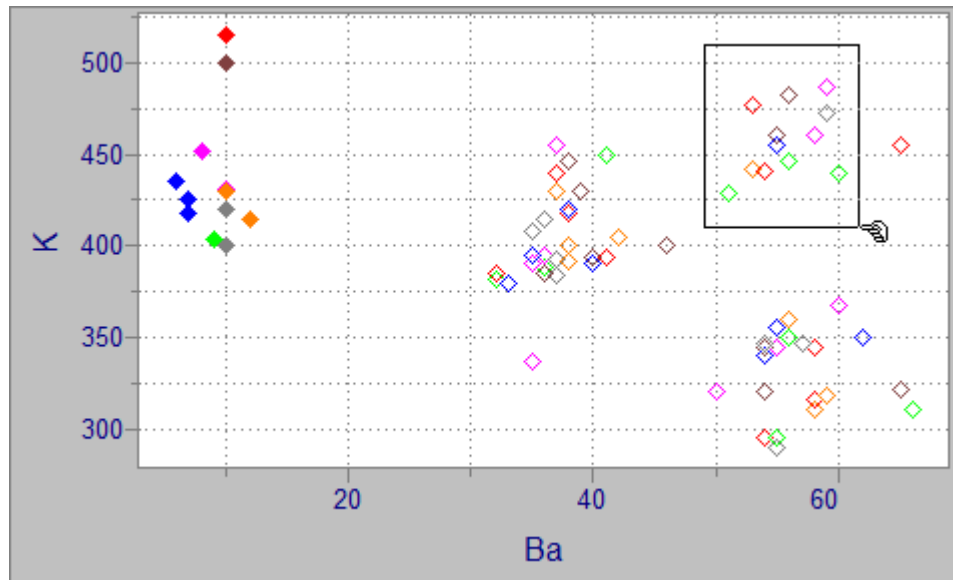
Figure 12.6
2D plot with some
points selected



These points remain highlighted until you deselect them. To deselect points, click the mouse button when the Pointer cursor is within the plot region. To select non-adjacent points in the plot or to deselect points in the middle of a region of selected points, use the


Ctrl key approach described in “Selecting in Graphics” on page 10-2. The next figure shows an example of making multiple selections in a 2D plot.

Figure 12.7
Making multiple
selections while
holding down the
Ctrl key



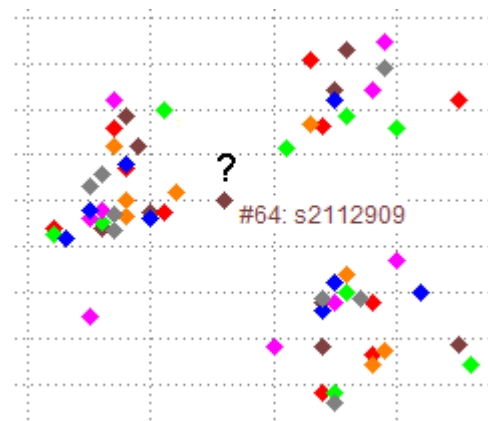
IDENTIFYING POINTS

When points in a scatter plot are unlabeled, the plots looks clean, especially when many samples are plotted. However, to display some point information,

- Click on the ID button  in the Ribbon
- Move the cursor to a point and press the left mouse button.


When the hot spot of the ID cursor is near a sample point, the row number and name of the point is displayed, as shown next.

Figure 12.8
Showing a row
number/name with
the ID tool



POINT LABELS

To turn labels on/off for all points in the active scatter plot,

- Click on the Label button  in the Ribbon

To override the preference set by Windows > Preferences > Chart > Label Attributes for the active scatter plot,

- Select Point Labels from the Display menu
- Select either None, Index, Name or Class from the submenu

CLOAKING

It is often helpful to highlight important points in a plot to focus on them. This becomes easier with cloaking, a process which hides selected or non-selected points. The cloaking tool is a three-way switch. Its three positions are:

- Show all points
- Show selected points
- Show unselected points

To see cloaking after selecting some points and making a scatter plot active,


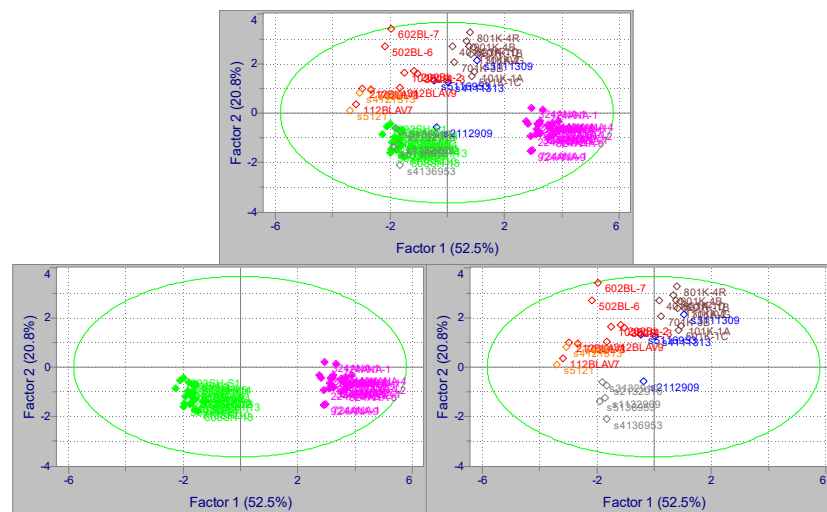
- Click on the Cloaking button  in the Ribbon


Figure 12.9 shows a 2D plot in which several points have been selected. Successive clicks on the cloaking button hide unselected and selected points.

Figure 12.9
A 2D plot with (a) no cloaking, all points shown; (b) only selected points shown, and (c) only unselected points shown



MAGNIFYING REGIONS

To magnify a portion of a plot,

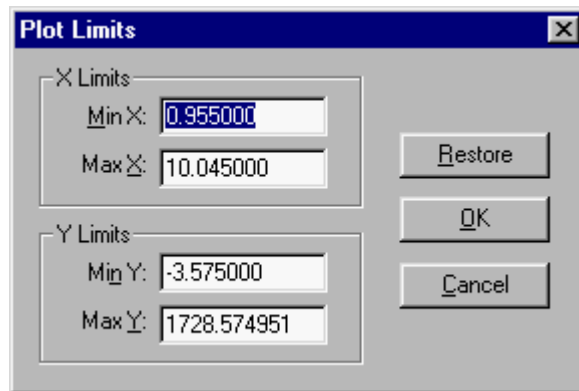
- Click on the Magnify button  in the Ribbon
- Position the magnifying glass over a spot in the plotting region
- Click-drag to define a rectangular area to enlarge
- Release the mouse button and the magnified view will be displayed

To unmagnify a plot,

- Click the right mouse button with the magnifying glass in the plotting region

It is also possible to set explicit plot axis limits using the Display > Limits menu item. The resulting dialog is shown below.

Figure 12.10 Display Limits dialog



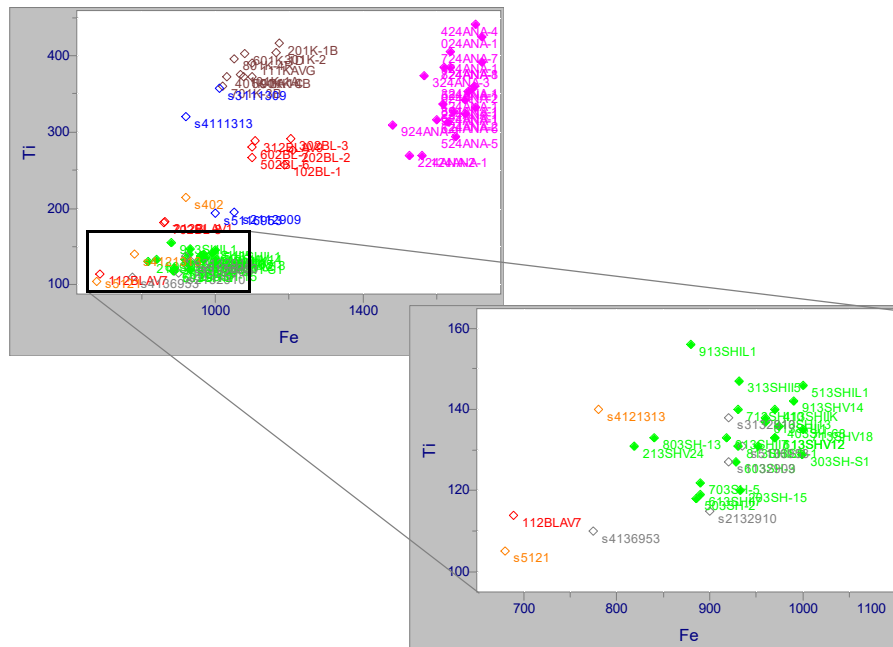
If you magnify a plot successively or change the display limits, you can unmagnify back through the reverse sequence with a series of right-mouse clicks. To immediately restore a successively magnified plot to its original view,

- Double-click the left mouse button

If you spin a magnified 3D plot, it reverts to the unmagnified state before spinning.


Magnifying a plot is shown in the following example.

Figure 12.11 Magnifying an area



SPINNING A 3D PLOT

A dynamic 3D scatter plot is an extremely useful visualization aid. To spin points in a 3D plot in the active window,

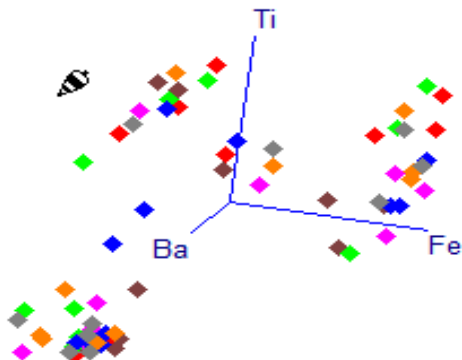
- Click on the Spinner button  in the Ribbon

When you move the cursor over the plot area, it takes on the form of a top. The cursor then controls the rotation of the points.

- Move the cursor to the upper left portion of the 3D plot

- Click and slowly drag the cursor across the plot area while holding down the mouse button, as shown below.

Figure 12.12
Rotating a plot with
the Spinner tool



The movement of the points corresponds to the cursor direction. Points remain selected in a 3D plot during spinning. If you move the spinner at a moderate speed across the plot and then release the mouse button while still moving, points continue to spin, a phenomenon termed *momentum spinning*. The velocity of rotation is a function of the speed of cursor movement when you release the mouse button. To end rotation, click once with the mouse button in the plot area.

Spinning with the Arrow Keys

Keyboard control of 3D rotation is provided via arrow key combinations. The combinations described below produce slow spinning. Pressing the Ctrl key in conjunction with the combinations increases the rate of rotation by 4 times.




Table 12.2
Spinning with
keyboard control

Keys	Spin Direction
Right Arrow	Left to right around Y-axis
Left Arrow	Right to left around Y-axis
Up Arrow	Bottom to top around X-axis
Down Arrow	Top to bottom around X-axis
Alt-Right Arrow	Clockwise around Z-axis
Alt-Left Arrow	Counter clockwise around Z-axis

Spinning with Spin Control Buttons

The Spin Control buttons, located on the far right of the Ribbon, operate in a similar fashion to the arrow keys but involve the mouse. Three pairs of buttons control the spin direction around each axes are described in the table below. Click on any button to spin points in the indicated direction.

Table 12.3
Spin Control buttons
in the ribbon

Tool	Description
	Spin around the X-axis of the view from top to bottom or from bottom to top
	Spin around the Y-axis of the view from left to right or from right to left
	Spin around the Z-axis of the view in a clockwise or counter clockwise direction

The effective rate of rotation increases with each successive click on the same button: the second click doubles the apparent spin rate over the initial rate, the third click triples the apparent spin rate, etc. To stop spinning begun with a ribbon button, click with the spin cursor in the plot area.

Rotation scheme

The rotation scheme is by default based on a cylindrical format: points rotate around the axis perpendicular to the direction of motion of the mouse cursor. Another form of rotation is based on a the rolling motion of a sphere, a method found more natural by some users. In this scheme, it is as if the points are inside a rolling sphere, and the cursor acts like a hand on the ball. Which rotation scheme is in use is governed by a View preference. [Figure 10.15, on page 10-12](#) shows how to change the setting.

Depth cueing

When many points are rotated in a 3D plot, it can be difficult to determine which points are “in front” of others. Pirouette offers an option to evoke depth by coloring points “behind” the plane of the screen differently than “in front”. If depth cueing is turned on (see [Figure 10.17, on page 10-13](#)), points behind the viewing plane are shaded gray. Labels also turn a gray color when a labelled point is behind the viewing plane.

PLOT SCALING

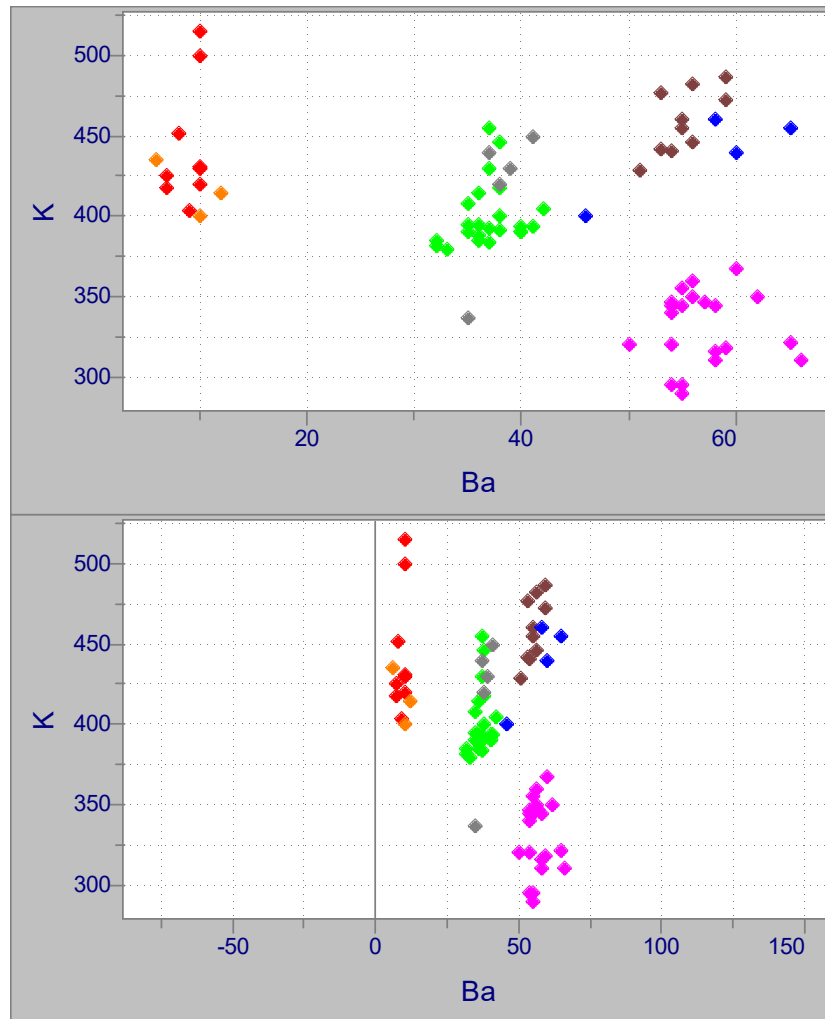
In scatter plots, two options affect the appearance of data.

Range vs. Data Scaling

The default setting for 2D and 3D plots in Pirouette is range scaling. Range scaling produces a plot in which sample points are distributed across the full extent of each coordinate direction in the plot area. However, range scaling may obscure relationships when the magnitudes of responses along the different axes are considerably different.

Data Scaling produces a plot with both axes having a range equal to that of the axis of largest range. By selecting the Data Scaling item in the Plot Scaling submenu of the Display menu, you convert a range-scaled plot to one in which points are presented proportional to the actual ranges in the data. An example of these two plot modes is presented below.

Figure 12.13
 Range scaling (top)
 Data scaling
 (bottom)

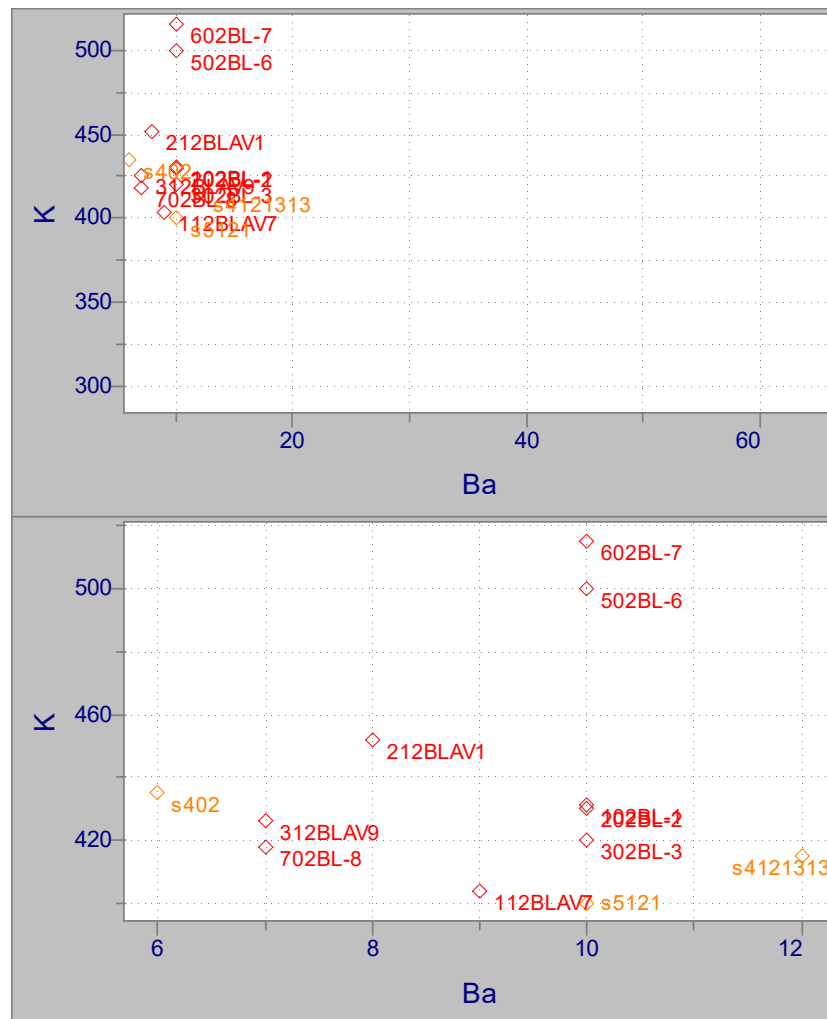


When a row index is on an axis or when the axes are associated with disparate quantities, the Data Scaling option has no effect on the plot appearance, *e.g.*, see [Figure 7.15](#), on [page 7-22](#).

All vs. Included Points

Initially, axes of subset scatter plots are scaled to the range of all points in the data set, whether or not they are included in the subset. In some cases, subset points occupy a small portion of the plot area, as shown in [Figure 12.14](#) below. To scale the axes to the range of points included only in the subset, select the Included Points item from the Plot Scaling submenu of the Display menu. The result of this action is shown in the second figure.

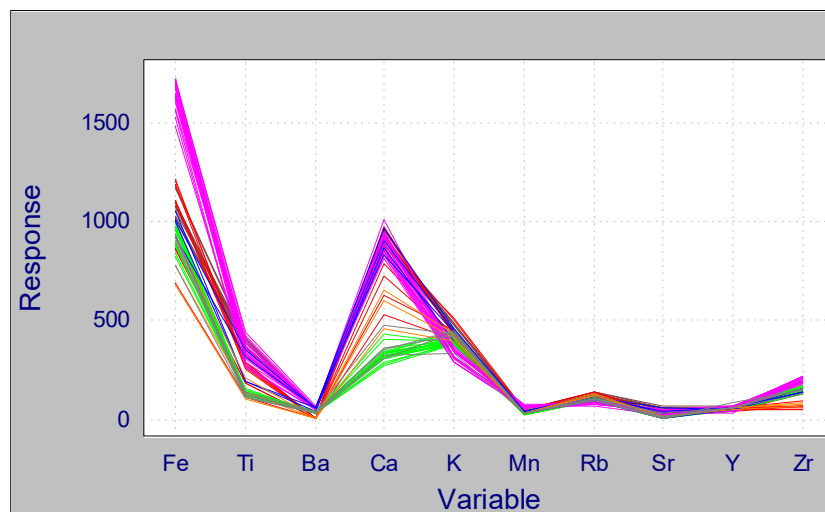
Figure 12.14
A subset scaled to all
points (top),
included points
(bottom)



Line Plots


Line plots contain either rows plotted against a column index or columns plotted against a row index. The points are connected to produce lines as shown in Figure 12.15 below.

Figure 12.15
A line plot of the
ARCH data



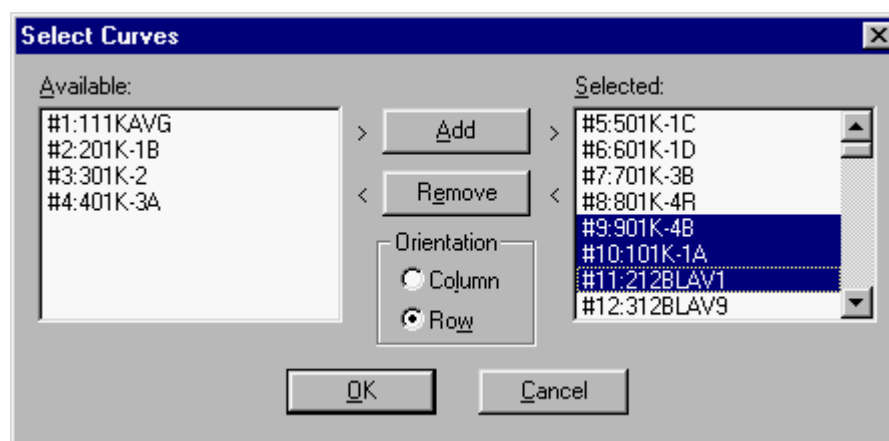
SPECIFYING AXES AND ORIENTATION

When a line plot is first displayed, its orientation (that is, row or column) depends on the object. For raw data, the default is to plot the rows as lines. For most computed objects, the default is to plot the columns as lines; but objects with the same size as the raw data (e.g., X Residuals) have rows plotted as lines. To change the orientation,

- Click on the Selector button  in the Ribbon

and the Selector dialog box shown in the figure below opens. Change the orientation by clicking either the Column or Row button; the appropriate items will be placed in the Selected list. For computed objects containing only one row or column the corresponding Orientation radio button is grayed.

Figure 12.16
Line Plot Selector
dialog



To add or remove lines, use the techniques outlined in “[Selecting in Lists and Tables](#)” on [page 10-1](#) to highlight entries in the right or left list. To select all entries in a list,

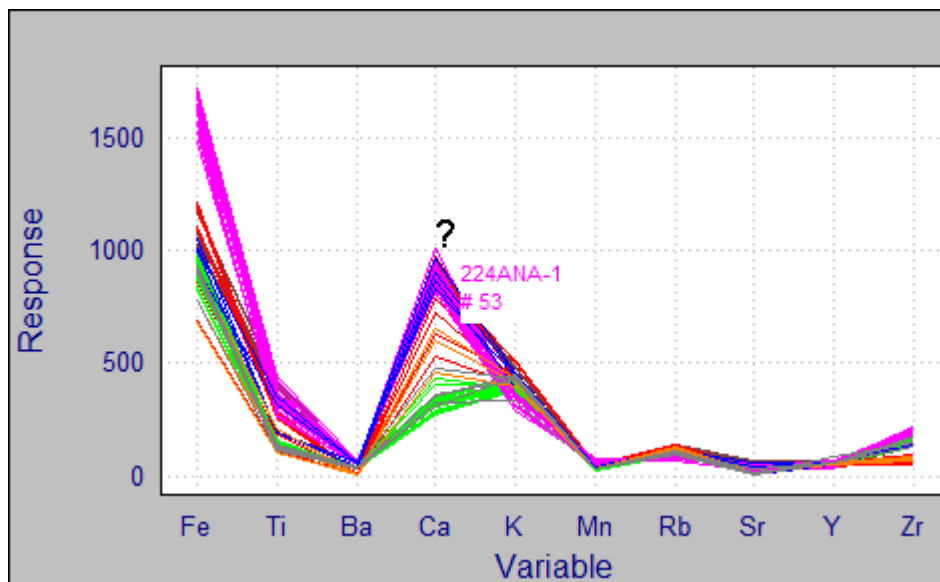
- Click on any entry
- Press Ctrl-A (for All)

Next, click on Remove or Add as appropriate, then on OK to produce the modified plot.

IDENTIFYING LINES

The ID tool acts in a line plot as it does in a scatter plot. See the discussion on [page 12-7](#) for additional details. The next figure shows an example of this behavior.

Figure 12.17
Using the ID tool in a
line plot



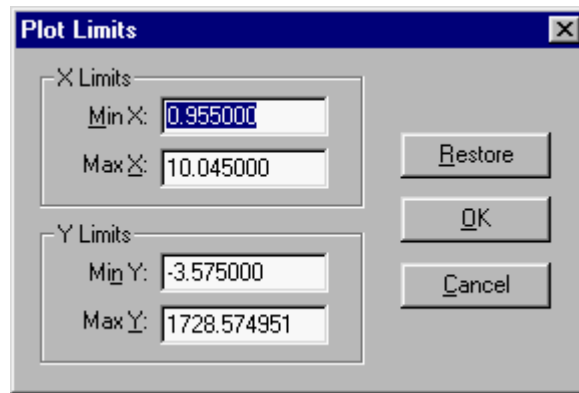
The name/number displayed corresponds to the trace with its vertex closest to the question mark. Click-drag across the various lines, and the displayed number/name changes.

Line color is determined by the color sequence described on [page 10-18](#). The number/name text shown with the ID tool (discussed below) has the same color as the line it identifies, which helps distinguish among adjacent lines.

MAGNIFYING REGIONS

The Magnifying tool works in the same fashion as described for a scatter plot on [page 12-8](#). It is also possible to set explicit plot axis limits using the Display > Limits menu item. The resulting dialog is shown below.

Figure 12.18
Display Limits dialog



Note: Note that the X Limits values are in variable index limits, not in the units that may be shown as the variable names.

PANNING LINE PLOTS

If a line plot has been magnified to show a restricted time region of the profiles and you want to show a different subregion, there are two procedures you can use:

- Unmagnify the region then select a new region to magnify, using the magnifier tool (see “[Magnifying Regions](#)” on page 12-8), or
- Pan the line plot to another region

To pan a line plot,

- Press and hold one of the arrows on the keypad, then click with the Magnify cursor.

The plot will pan in the direction of the arrow key by an amount equal to the width or height of the current view. Thus, with the right arrow held, each click of the cursor will pan the plot to the right; with the left arrow held, each click of the cursor will pan to the left. When the edge of the plot view is reached, additional clicks will have no effect.

To 'unpan', just as in unmagnify, do a right-click with the magnifier cursor to go back to the previous view.

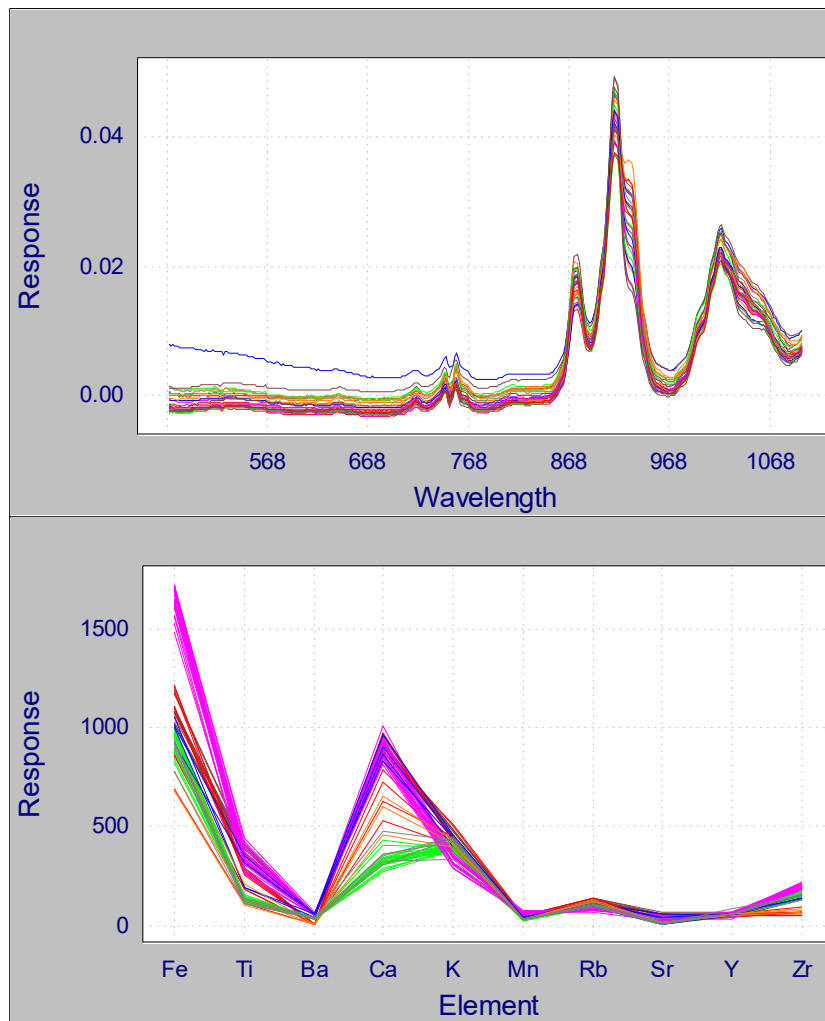
AXIS LABELS

The default labeling of the X axis for line plots is determined by a Preference discussed in “[Axis Labels](#)” on page 10-17. To override this setting,

- Select Axis Labels from the Display menu
- Choose the available entry (either Name or Number)

Line plots with custom labels entered in the Plot Label Attributes dialog are shown in the following figures.

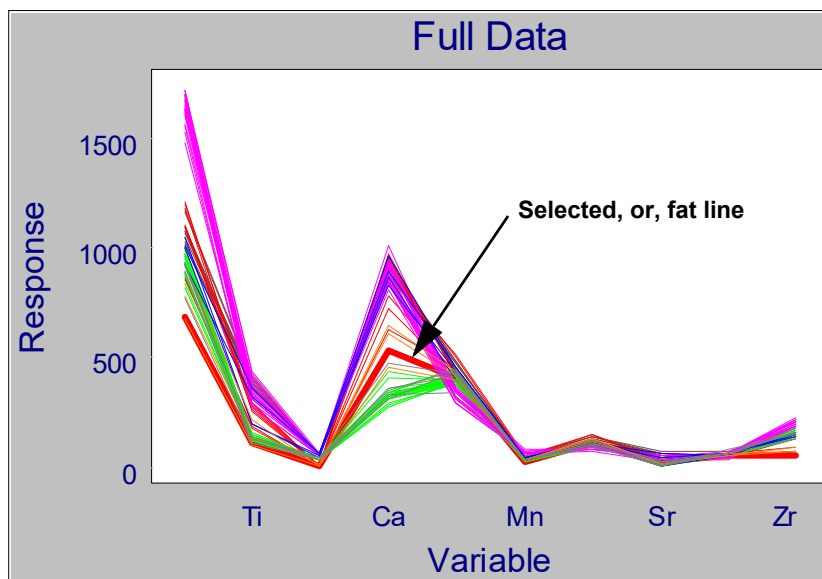
Figure 12.19
Custom X-axis name
labels with
continuous variables
(top), discrete
variables (bottom)



SELECTING LINES

Selecting lines in a line plot is accomplished in the same fashion as described earlier for a scatter plot. See the discussion on [page 12-5](#) for additional details. A line plot with a single selected (*i.e.*, fat) trace is shown below.

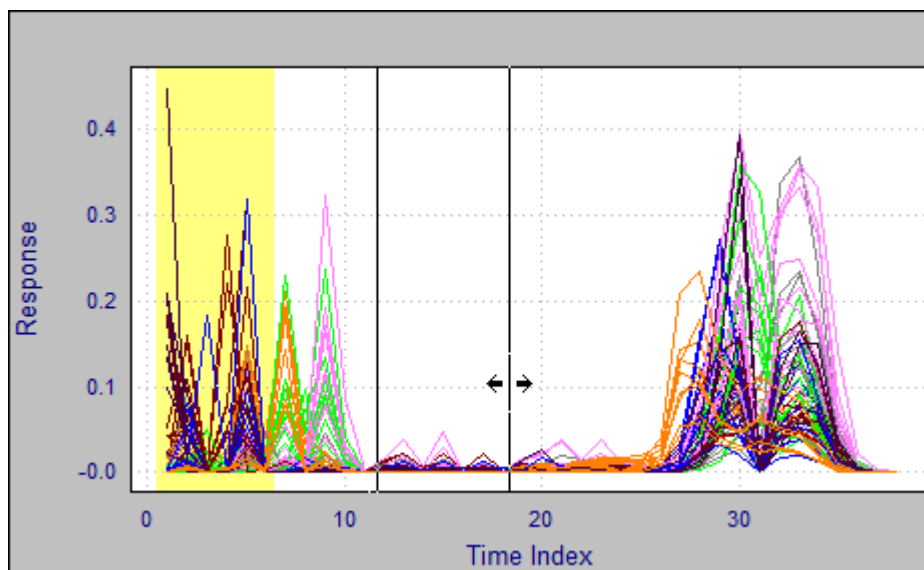
Figure 12.20
A selected trace in a
line plot



SELECTING RANGES

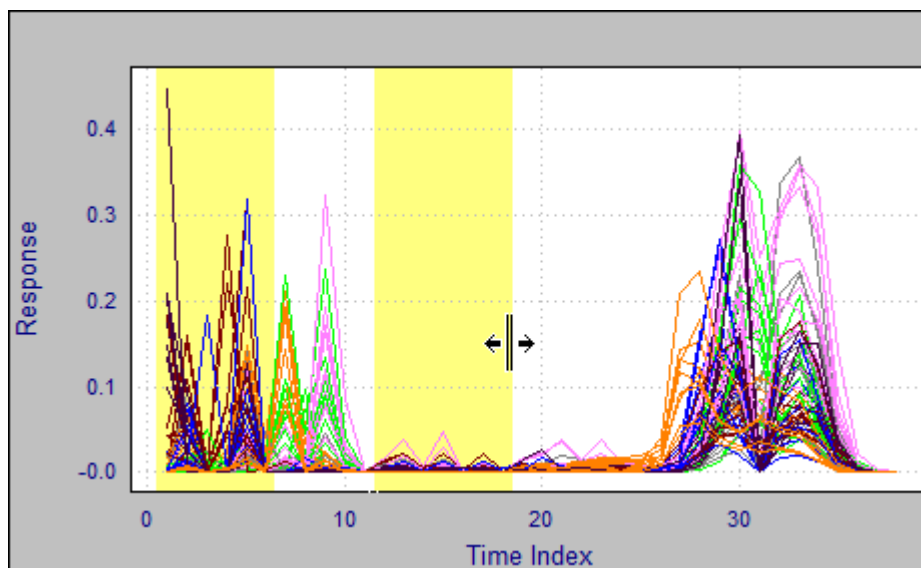
With continuous variables (*i.e.*, in spectra or whole chromatograms) it is often useful to select regions using the Range tool. To select discontinuous regions, Ctrl-click-drag. A line plot with multiple selected regions is shown below.

Figure 12.21
Selecting ranges in a
line plot



When selecting a new range, the range tool shows a single vertical bar, such as in the previous figure. If the tool is hovered over the edge of an existing selected range, as shown in Figure 12.22,


Figure 12.22
Using Range Tool to
extend a selection



it shows a double vertical line. This is an indication that if you click-drag the tool from this location, you will extend the existing selection.

REDRAWING TRACES

Normally traces are drawn in order of the colors of the lines, a fast way to draw when many lines (*i.e.*, samples) are plotted. However, such an approach may obscure subtle details in lines whose colors get drawn first. To see traces drawn in the order they appear in the data table,

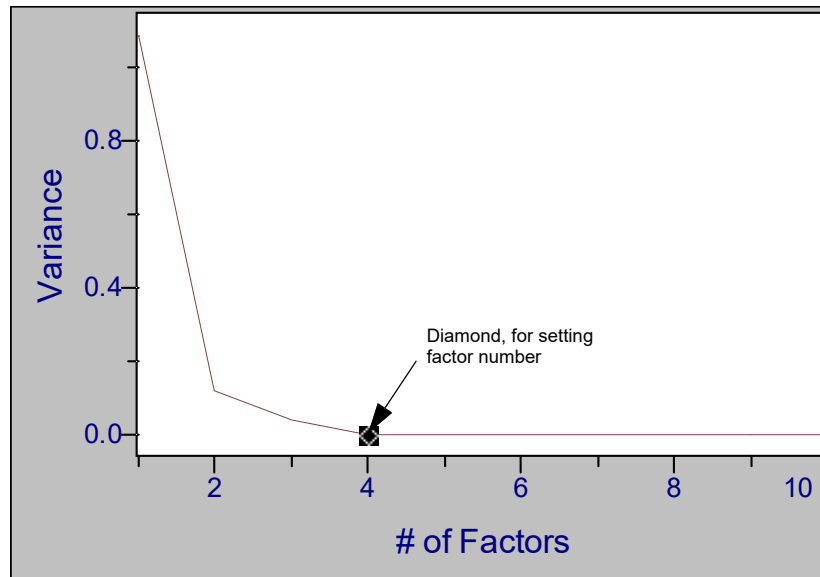
- Click on the Redraw button  in the Ribbon

This feature is useful when, for example, row number corresponds to the order in which samples were collected. The rate at which the traces are displayed is determined by the Windows > Preferences > View > Line Plot > Redraw Delay preference shown in [Figure 10.19](#).

FACTOR SELECTION LINE PLOTS

A special type of line plot allows the user to change a setting sometimes described as the optimal (or best) factor. CLS, PCA, KNN, SIMCA, PLS, PLS-DA, PCR and ALS all produce this type of line plot which can be converted only to a table view. Factor selection line plots have a diamond whose position is changed by clicking the mouse along the trace. The current x axis setting (a positive integer) of the diamond in a factor selection line plot is displayed in the lower right corner of all 2D and line plots of algorithm results dependent on the setting.



Figure 12.23
PCA factor selection plot



Multiplots

A multiplot is a special collection of 2D scatter plots. Any object which contains more than two columns/rows can be viewed as a multiplot. Multiplots must be zoomed before user interaction can occur. The following table summarizes the various zoom/unzoom possibilities.


Table 12.4
Zooming and Unzooming

	Zoom	Unzoom
Mouse	Double-click	Shift-double-click
Display Menu	Zoom Current Plot	Unzoom Current Plot
Keyboard	Enter	Ctrl-Enter
Button		

Note: *Zooming and Unzooming also apply to array plots, which are collections of subplots contained in a single window.*

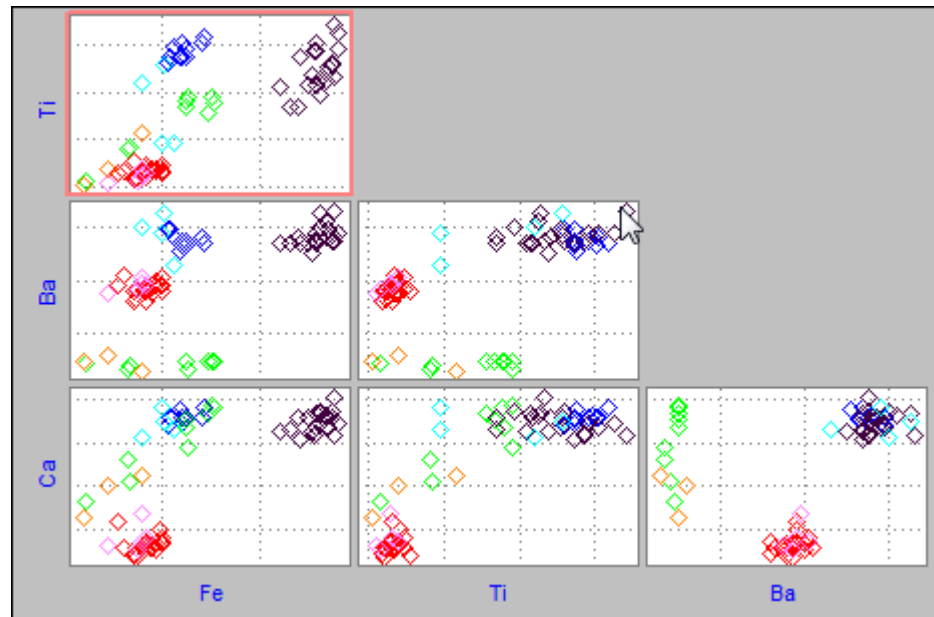
The subplot to be zoomed has a thick border called the *highlight*. The highlight can be moved by mouse clicking or by pressing the Tab key or the keypad arrow keys. Its color is set in the Multiplot View preference.

To convert an object with at least three columns/rows to a multiplot view,


- Click on the Multiplot button  in the Ribbon

A triangular display of variable-by-variable biplots, each a 2D scatter plot, is presented.

Figure 12.24
ARCH data multiplot
showing default
number of subplots

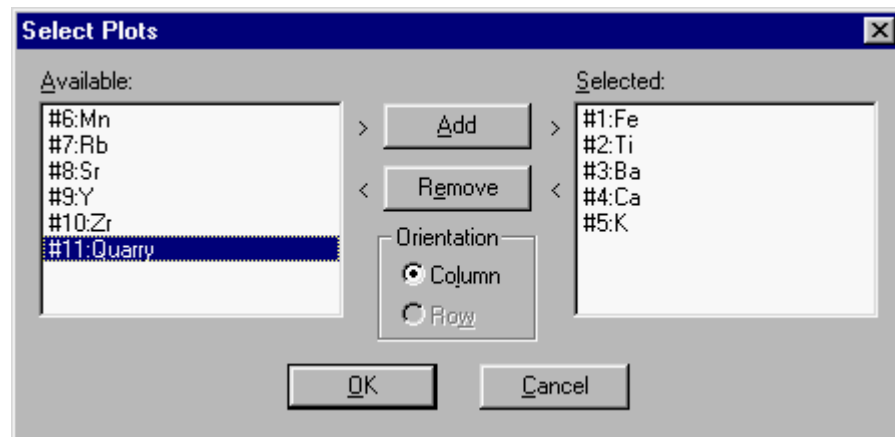


When a multiplot is first shown, it uses the first n variables, where n is the value set in the Multiplot View Subplot Columns preference. To change this default value, see [Figure 10.24](#), on page 10-15. To choose different variables for a particular plot,

- Click on the Selector button  in the Ribbon

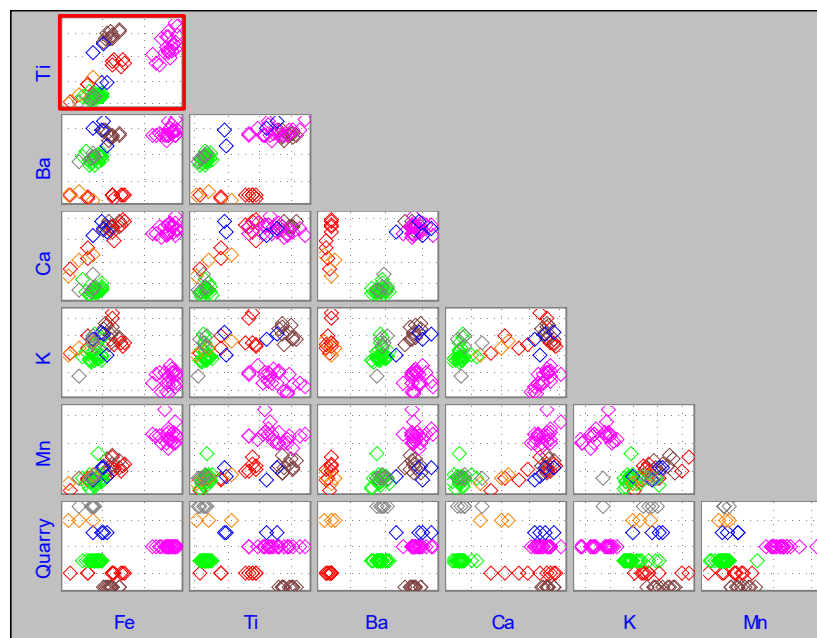
Under Selected are the names of the currently displayed columns; add or remove entries in the fashion described previously.

Figure 12.25
Selector dialog box
for a multiplot



The following figure shows the result of adding variables to the multiplot of [Figure 12.24](#).

Figure 12.26
Multiplot after
variables added



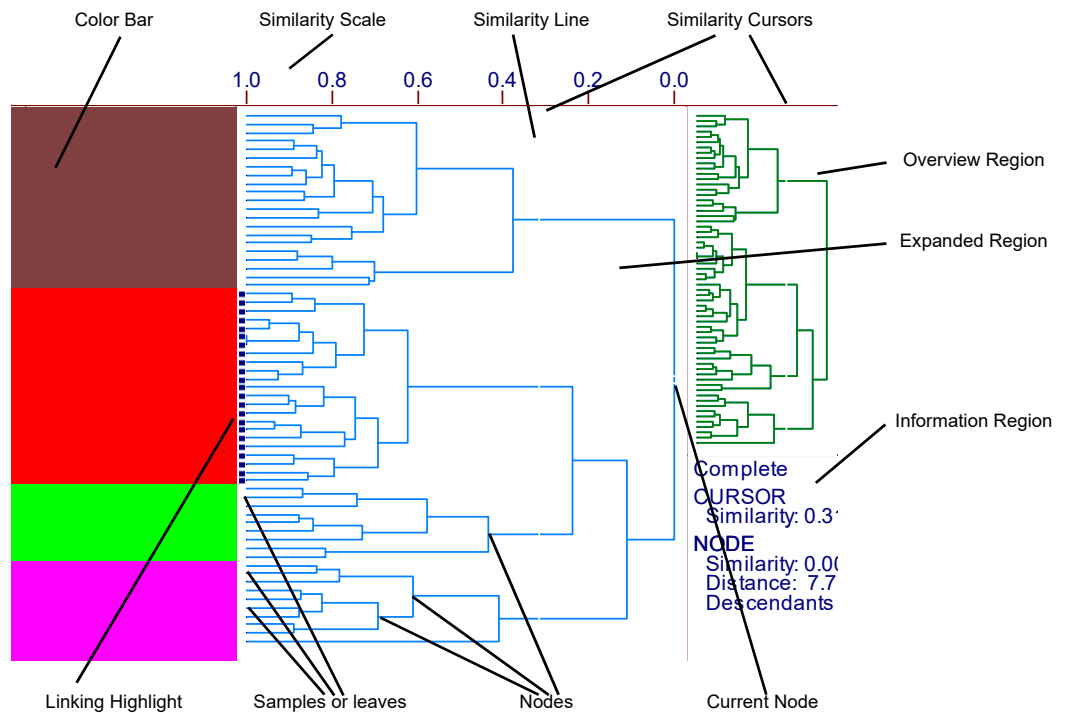
The Dendrogram

The HCA algorithm produces a single result, the dendrogram. Dendrograms provide a great deal of useful information by showing how samples (or variables) group and by pinpointing anomalous samples (or variables). Pirouette dendrograms are highly interactive, letting you examine differences in similarity values and focus on individual branches of the tree-like structure. What follows is a description of the dendrogram and how to interact with it, including how to set similarity values, identify leaves and create class variables. For a discussion of the HCA algorithm and an explanation of how various clustering techniques affect the dendrogram, see [“Hierarchical Cluster Analysis” on page 5-1](#). As is the case in that discussion, the term *sample* will be used when either sample or variable is meant.

THE DENDROGRAM ENVIRONMENT

As shown in [Figure 12.27](#), the dendrogram window is divided into four distinct screen areas: the color bar, the expanded region, the overview region and the information region. The dendrogram itself is always displayed as a tree growing from the right, dividing into more and more branches which eventually terminate on the left in individual samples, called leaves. The point where a branch divides is called a node. Non-leaf nodes always have two descendants which continue branching in pairs until they terminate in leaves. The dendrogram is displayed sideways for convenient leaf labeling.

Figure 12.27
Dendrogram window features

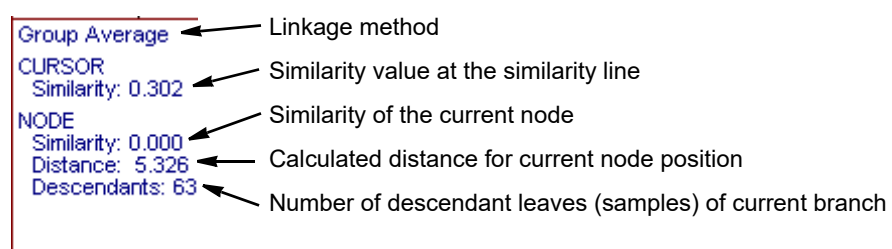


The dendrogram is scaled to similarity, a metric derived from relative Euclidean distances among samples. The similarity value at the tree root is 0.00. Terminal nodes, the individual samples, have a similarity value of 1.00. All nodes between the root and leaves are branches with similarity values between 0.00 and 1.00. For the definition of similarity and more information on the mathematics behind the dendrogram and Hierarchical Cluster Analysis, see [“Mathematical Background” on page 5-2](#).

Information Region

The following figure shows the information region of a dendrogram. The cursor and node values depend on the current location of the similarity line and the current node, respectively.

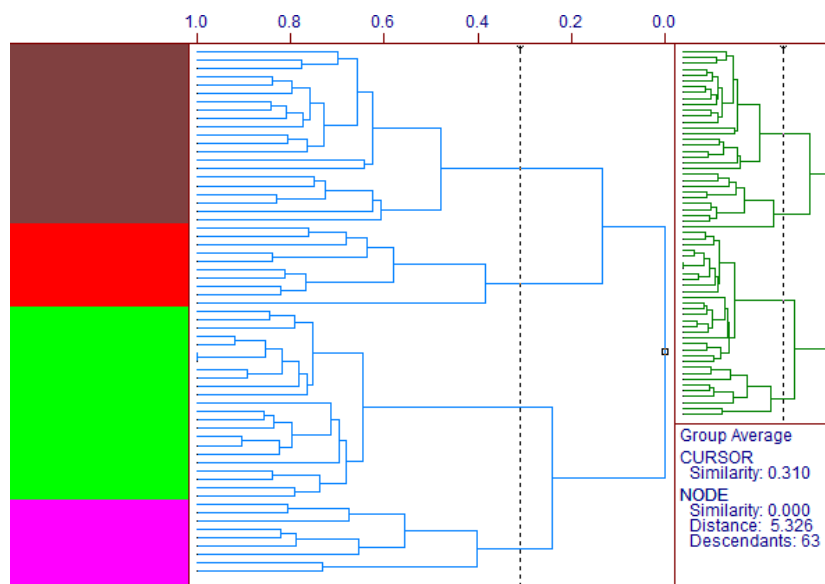
Figure 12.28
The dendrogram information region



Color Bar

The Color Bar occupies the left side of the dendrogram window and displays in color the effects of different similarity settings. The color bar is an aid to interpreting relationships among samples, based on clusters defined by a similarity threshold. When you move the similarity cursor to the left, then release the mouse button, the colors in the bar reflect the clusters defined by that similarity value; see [“Setting Similarity Values” on page 12-25](#).

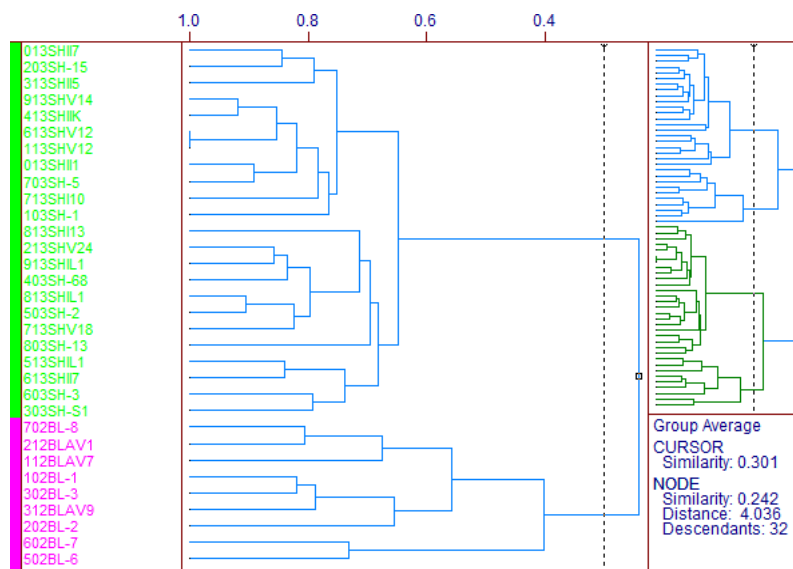
Figure 12.29
Dendrogram color bar



Overview and Expanded Regions

The overview and expanded regions are very similar and are therefore described together. The overview region, located in the upper right of the window, displays the entire dendrogram without labels. The overview region, a miniature of the complete dendrogram, is an orientation aid. If you zoom the dendrogram, the zoomed portion in the overview region is displayed in a different color as shown below. This color difference can be extremely helpful in a complex dendrogram when all levels of the expanded region look similar.

Figure 12.30
Zoomed region of a dendrogram showing a different color for the overview



The expanded region, the site of most interaction, shows the current subtree (which may be the entire tree). Here you can display new subtrees, select new nodes and change the color bar by moving the similarity line. If the sample spacing in view allows, the color bar will be replaced by the sample names as shown above.

DENDROGRAM NAVIGATION

Interaction possibilities in the dendrogram are rich. You can zoom into and out of subtrees and even create new class variables based on user specified similarity values.

Mouse navigation

- Click on a node in either the overview or expanded region to check similarity values and distances at that node.
- Double-click on any node in either region to fill the expanded region with nodes to the left of the selected node.
- Double-click on the far right of the of the dendrogram in the overview region to restore the complete tree to the expanded region.
- Point at the right edge of the expanded region. When the pointer becomes a right arrow, click to move one node up the tree. (Nothing happens if the dendrogram is full size.)

Keyboard navigation

- Up-arrow moves to the upper descendant node, equivalent to moving toward the left and top of the screen.
- Down-arrow moves to the lower descendant node, equivalent to moving toward the left and bottom of the screen.
- Right-arrow moves to the ancestor node of the current node (up a level), toward the right side of the screen.
- Pressing Enter zooms the expanded region to the current node position.

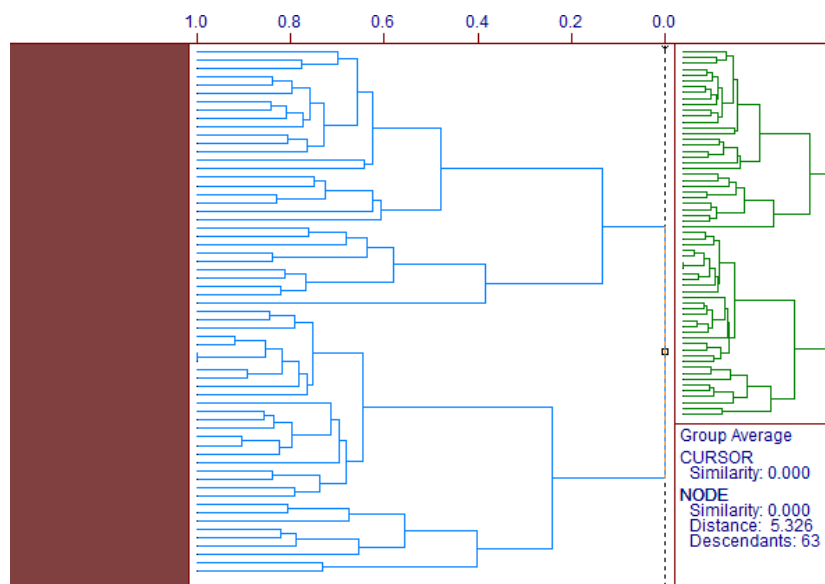
SETTING SIMILARITY VALUES

At the top of both the overview and expanded regions is a small handle that looks like an inverted caret symbol. The handle is attached to a vertical dotted line which appears in both regions. When the cursor is moved over the handle, it becomes a horizontal, double-headed arrow. Click-drag this arrow to move the similarity line and modify the similarity value setting. The similarity line, which establishes clusters, can only be moved with the mouse.

When a dendrogram is first generated, the similarity line is placed initially at a value of 0.0. Such a dendrogram is shown in [Figure 12.31](#). Consequently, the color bar along the left is a single color. Moving the similarity cursor to the left until it crosses more than one branch causes the color bar to take on more than one color—each cluster to the left of the similarity threshold is colored differently.

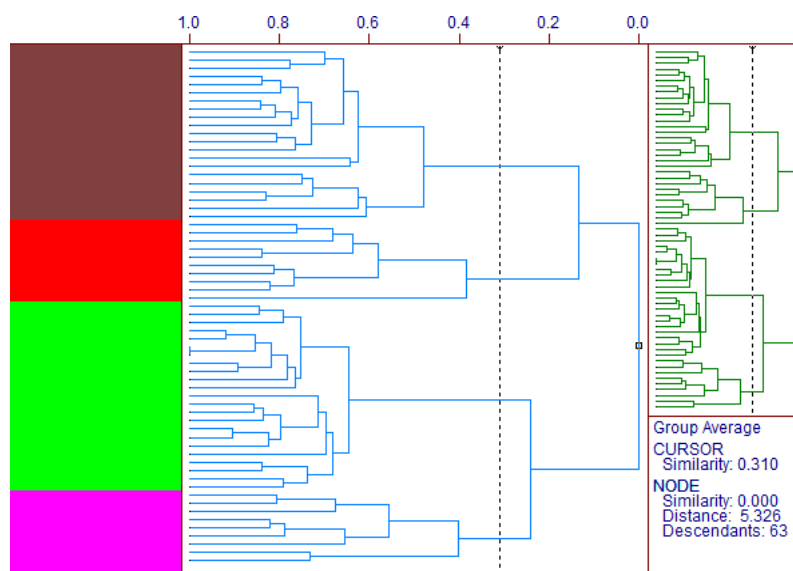
Colors are assigned according to the color sequence; customizing the color sequence is described in [“Color Sequence” on page 10-18](#). The color bar initially takes on the first color in the sequence. As the similarity cursor is moved to the left and the bar breaks into colors, colors are assigned in the sequence order. If more branches are broken than there are colors in the sequence, the color pattern repeats with the first color in the sequence for the next branch cluster. Thus, more than one cluster may have the same color.

Figure 12.31
A dendrogram with
similarity line at 0



The following figure shows the dendrogram of [Figure 12.31](#) after the similarity line has been moved to the left. Note the four colored clusters.

Figure 12.32
A dendrogram with
four colored clusters

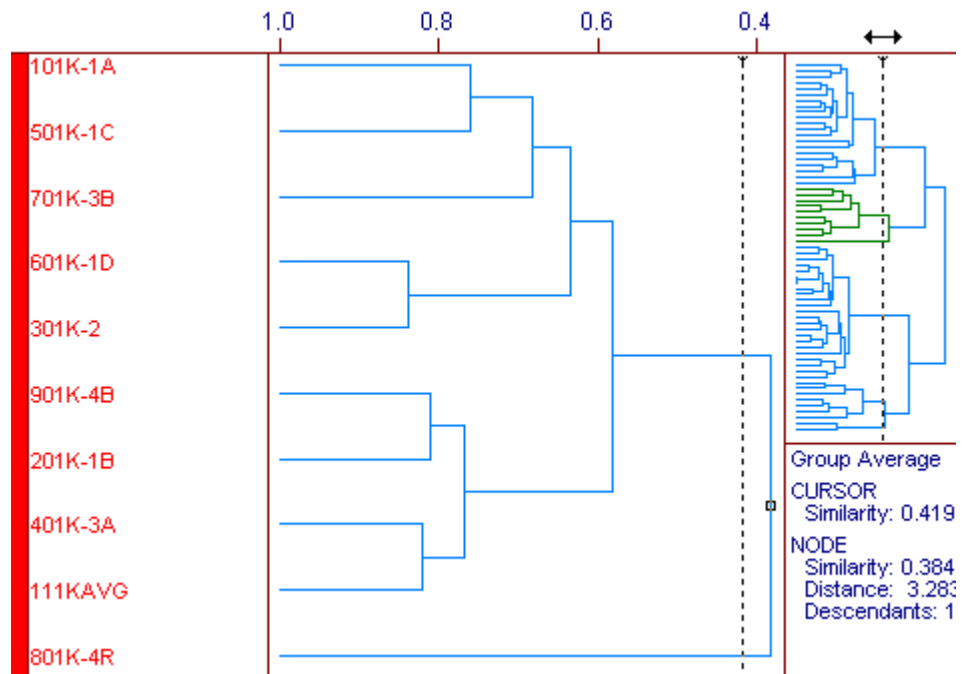


In a zoomed expanded view, the similarity line may no longer be visible. In this case, find the similarity line in the overview to the right of the highlighted cluster. To move the similarity line into the expanded region,

- Grab the similarity line in the overview region and move it to the left

Eventually it will appear in the expanded region, as shown below. Grab it in the expanded view for finer control of the similarity line position.

Figure 12.33
Moving the similarity
line in the overview



CREATING CLASS VARIABLES

If a dendrogram exhibits obvious sample groupings, they are a natural basis for categorizing the samples. A class variable can be constructed by assigning each sample within a cluster (a region of the same color in the dendrogram) the same class value with different clusters having different values. This feature is particularly useful for classification problems which lack predefined class assignments. Of course, assignments derived from a dendrogram must always be viewed with some skepticism until their validity is confirmed.

To create a class variable automatically from a sample dendrogram in the active window,

- Select Activate Class from the Edit menu

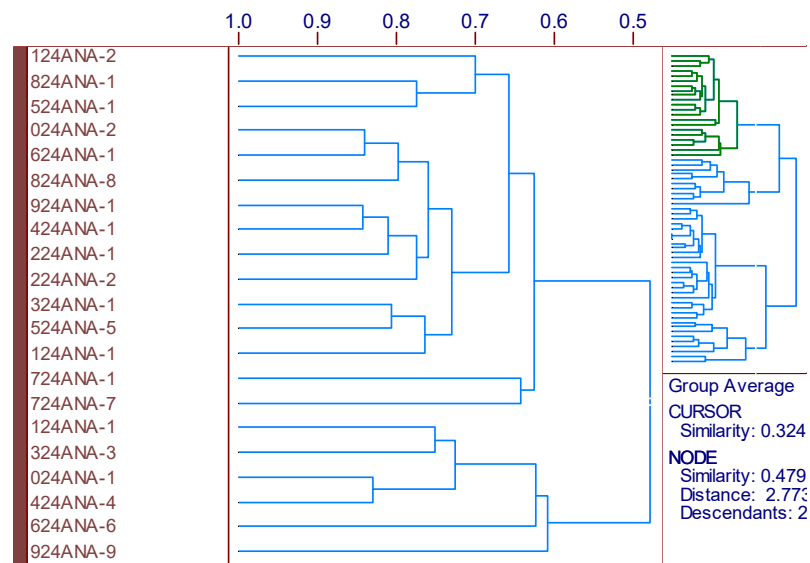
Activating a class from the dendrogram accomplishes two things. First, a new class variable is created based on dendrogram clusters. You can see this new variable in the class variable region of the Full Data table view. Because you can create more than one class variable from the dendrogram, a naming convention has been established which allows the user to distinguish the various instances. The names are composed of the first 6 characters of the subset name plus the first two digits of the similarity value when the class was activated. Thus, when the dendrogram shown in [Figure 12.32](#) is the active window, choosing Activate Class produces a class variable named QUARRY31.

The second action which occurs upon selecting Activate Class is that the newly-created class is activated, *i.e.*, the class values are used to color the data points in any appropriate scatter or line plots; see “Linking Views” on [page 12-28](#). The active class name appears at the bottom of the Pirouette window as a reminder. By activating a class and thereby mapping cluster color to sample points in other plots, you can compare clustering in the dendrogram to that found in other sample oriented objects, *e.g.*, PCA scores plots. This may allow you to come to some conclusion concerning the validity of clusters defined by the dendrogram.

IDENTIFYING SAMPLES

The initial view of the dendrogram may show a solid color bar or names, depending on the size of the window and the number of leaves displayed. If the color bar is solid and you zoom a branch of the dendrogram with fewer leaves, names will appear if you zoom far enough. The following figure shows a zoomed dendrogram with sample names. Names take on the same color as that of the cluster(s) containing that leaf. Not only can you see the clustering of samples through the placement of the similarity line, but by zooming the dendrogram view, you can also compare the names of clustering samples.

Figure 12.34
Zoomed dendrogram
showing sample
names



To identify a single sample when the Color Bar is solid, use the ID tool as described in “Identifying Points” on page 12-7. In this case, place the hot spot of the question mark over the leaf you wish to identify.

Linking Views

Linking is a visualization feature which enhances Pirouette’s graphical approach, permitting a more complete investigation of the data. Linking is the process of highlighting samples (or variables) in one object view and showing that highlight status in views of related objects. Linking gives you different perspectives from which to evaluate the relationships among samples (or variables). Linking can be initiated from any sample or variable oriented object view which allows highlighting. This includes table and scatter plot views of raw data and algorithm results as well as the dendrogram. You can link either variables or samples or both, though this last option can become confusing.

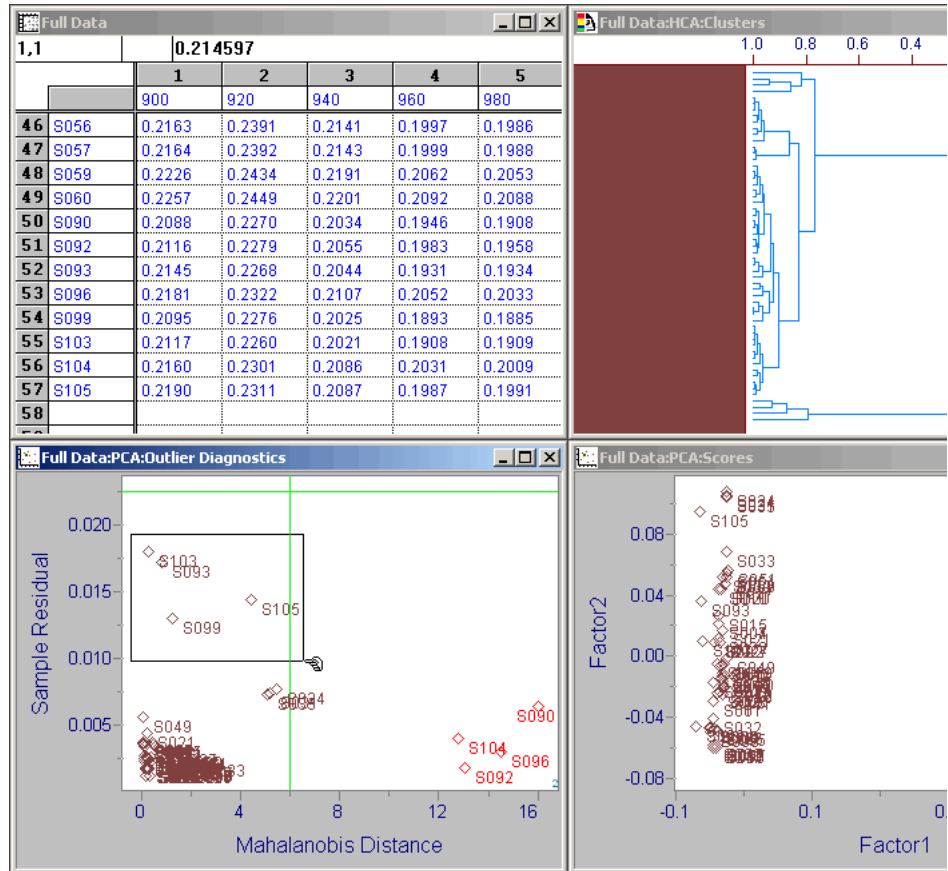
To highlight points in a scatter plot,

- Click on the Pointer button
- Click–drag around a region containing points

and a “rubber box” becomes visible and defines the region. Upon releasing the mouse button, enclosed point(s) are highlighted. The Pointer tool works in much the same way to select samples from a dendrogram.

The following figure shows several sample oriented objects which have no linked selections: two PCA results, a dendrogram and a table view of Full Data. However, the linking is about to be established from the Outlier Diagnostic result; the rubber box is encompassing four points.

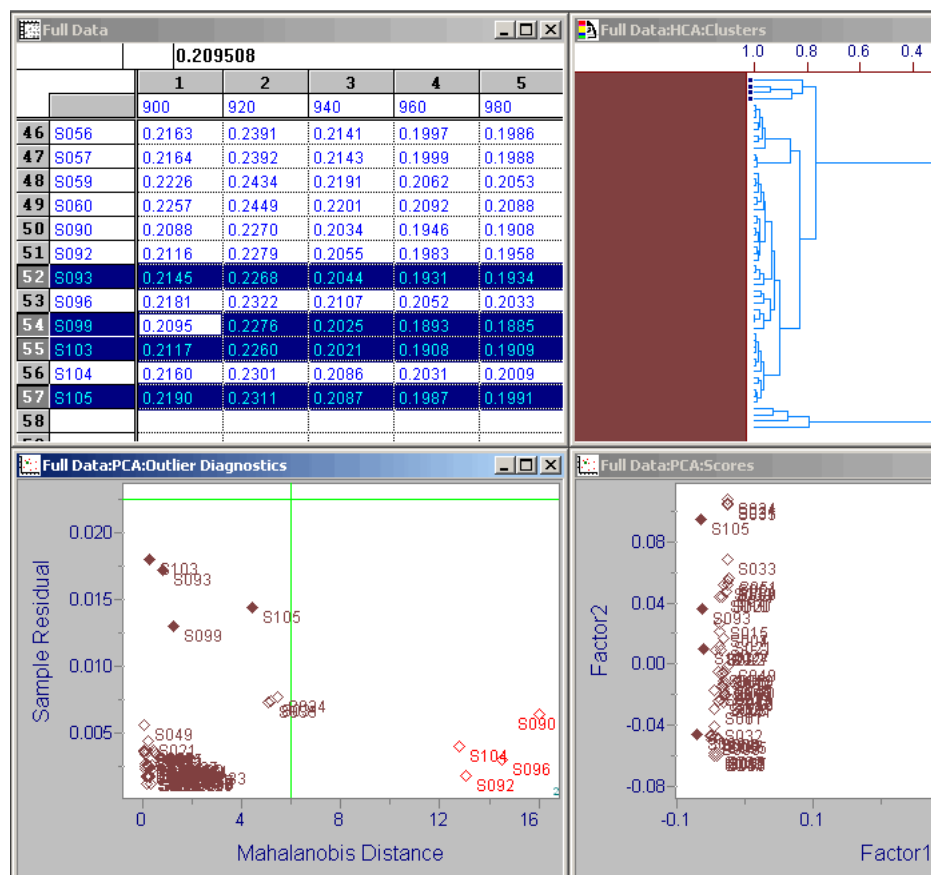
Figure 12.35
Sample-oriented views before highlighting



The next figure shows the same set of objects after linking of selections has been established. Note the top leaves with the samples highlighted in the dendrogram. The rows of Full Data show the selected state of the samples—the same four samples are highlighted—and the same samples also appear highlighted in the Scores and Outlier Diagnostic plots at the bottom of the figure.

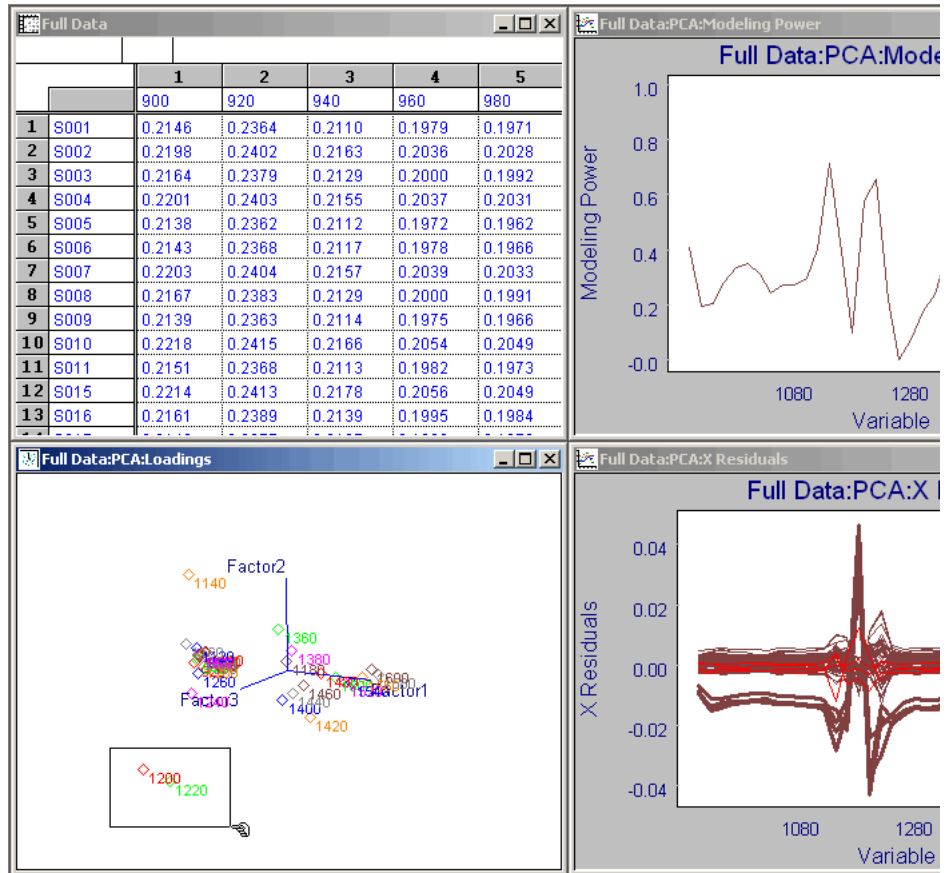
12 Charts: Linking Views

Figure 12.36
Sample-oriented
views after
highlighting



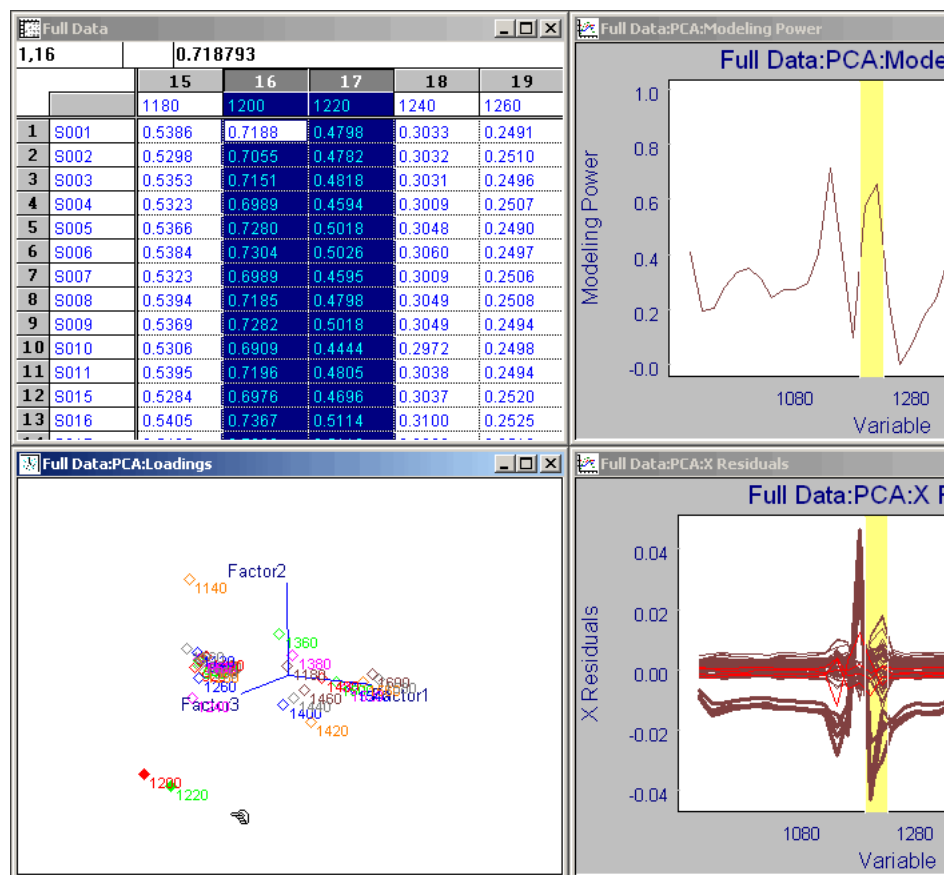
So far we have emphasized sample linking. However, some objects are variable-oriented, such as the PCA Loadings. No change will occur in variable-oriented graphics when a sample is highlighted in a sample-oriented graphic, but highlighting the variables in a variable-oriented plot **will** link those selections to any other variable-oriented view. The figure below contains a collection of unlinked variable oriented objects: Loadings, X Residuals, Modeling Power and again, Full Data.

Figure 12.37
Variable-oriented
views before
highlighting



The next figure shows the same objects after two variables have been highlighted in the 3D loadings plot. The Pointer indicates the selected points. Note that the same variables are also highlighted in the table and line plot views.

Figure 12.38
Variable-oriented
views after
highlighting



Creating Subsets from a Graphic

Subsets provide a powerful approach to realizing what-if scenarios in Pirouette. Because subset creation requires highlighting, subsets can be created from any view which shows linking, *i.e.*, scatter plot and table views of sample and variable oriented objects and the dendrogram. Creating subsets from tables is described on [page 13-20](#).

To create subsets from a scatter plot or dendrogram,

- Select points with the Pointer
- Choose Create Exclude from the Edit menu

To create subsets from a line plot,

- Select traces with the Pointer and/or ranges with the Range tool
- Choose Create Exclude from the Edit menu

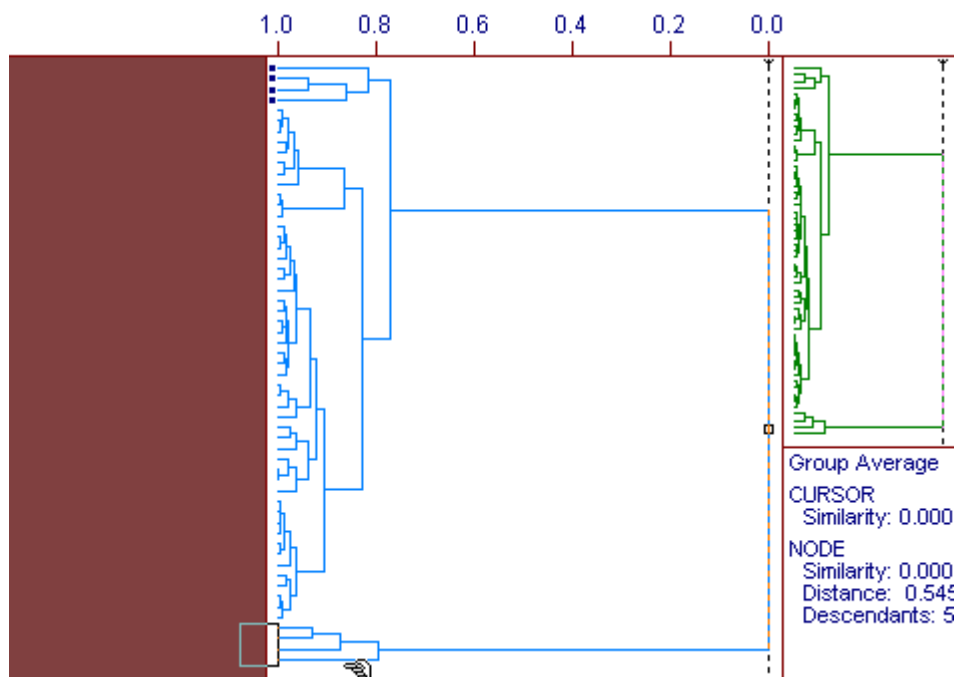
Both actions generate a new subset called Unnamed if it is the first exclusion set created, Unnamed-2 if it is the second, *etc.* If the graphic from which the subset is created is an algorithm result, that graphic continues to be displayed unaltered as the active window, and the newly created subset is found as a new entry in the Object Manager. To work with this subset, you must drag it from the Object Manager onto the Pirouette work area.

You can use the above methods to exclude samples and/or variables, depending on the orientation of the graphic. Thus, referring to [Figure 12.36](#), choosing Create Exclude would exclude the four highlighted samples while in [Figure 12.38](#), this action would exclude the highlighted variables.

Subsets can also be created from the dendrogram, as shown in [Figure 12.39](#), where the samples to be excluded are in the process of being selected. The arrow cursor changes to the Pointer when it enters the color bar region of the dendrogram, indicating that selecting can occur.

Note: *Creating subsets by exclusion is a quick way to remove outliers before (re)running an algorithm.*

Figure 12.39
Excluding samples
from a dendrogram
view



All of the above scenarios involve algorithm results. If, however, the 2D or 3D plot from which the subset is created is not an algorithm result, rather it is of the raw data, the selected points disappear when Create Exclude is chosen, and the window title changes to the new subset name. This is analogous to the case when a subset is created from a spreadsheet, *i.e.*, the new set is displayed in the active window.

Sometimes creating subsets from a 2D plot is more efficient than operating on the table view. For example, [Figure 12.40](#) contains two copies of a 2D plot of Full Data with a class variable on the y axis and sample # on the x axis, a combination which clearly distinguishes samples by category. (The two copies were made by composing one plot, clicking on the Drop button in the ribbon and dropping a second copy onto the workspace, then tiling the windows.) To break Full Data into two subsets based on category,

- Select samples to go in one subset (as shown below)
- Select Create Exclude from the Edit menu

Figure 12.40
Creating a subset from a scatter plot

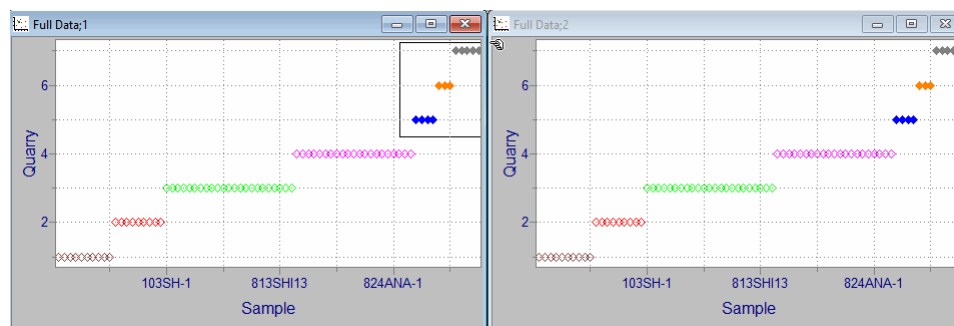
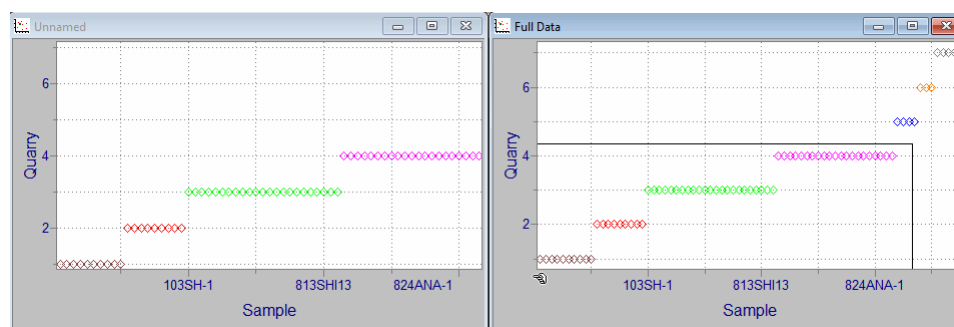


Figure 12.41 shows the result of this action. Note that the excluded points have disappeared from the left plot and its title has changed. To create a second subset containing the remaining samples,

- Click on the 2D scatter plot on the right
- Select the other samples (as shown below)
- Select Create Exclude from the Edit menu

Figure 12.41
Creating a second subset from a scatter plot



Plot Colors

Pirouette maps samples or variables to the same color across scatter and line plots, *i.e.*, the color applied to a particular sample is the same in all plots containing that sample. The colors assigned to points or lines come from the Color Sequence; see “Color Sequence” on page 10-18. Thus, sample #1 takes on the first color of the color sequence, sample #2 takes on the second color, and so on. If there are more samples than colors in the sequence, the mapping wraps around to the beginning of the sequence. So with 7 colors in the sequence, sample #8 has the same color as sample #1.

Note: The Color Sequence is also used to assign colors to sample and variable clusters in the dendrogram, based on the location of the similarity line. However, this form of color mapping is not analogous to that described above for line and scatter plots.

Pirouette provides another means of mapping colors to sample oriented plots. Because a class variable column can contain only integers, its entries can map to the color sequence in sample oriented scatter and line plots. This mapping is invoked by activating a class. To activate a class,

- Move to the class variable region in a table view of a subset
- Click on a class variable column head
- Select Activate Class from the Edit menu

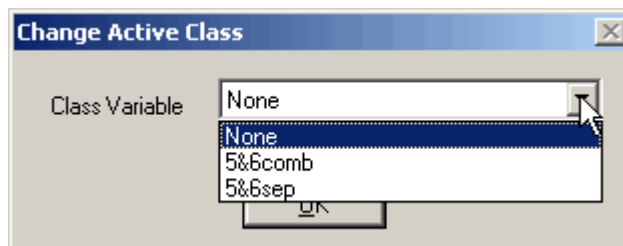
Alternatively, you can use the Activate Class button in the status bar, next to where the name of the current active class is displayed.

Figure 12.42
Activate Class
button in Status Bar



- Click the button to display the Active Class selection dialog.

Figure 12.43
Active Class selector



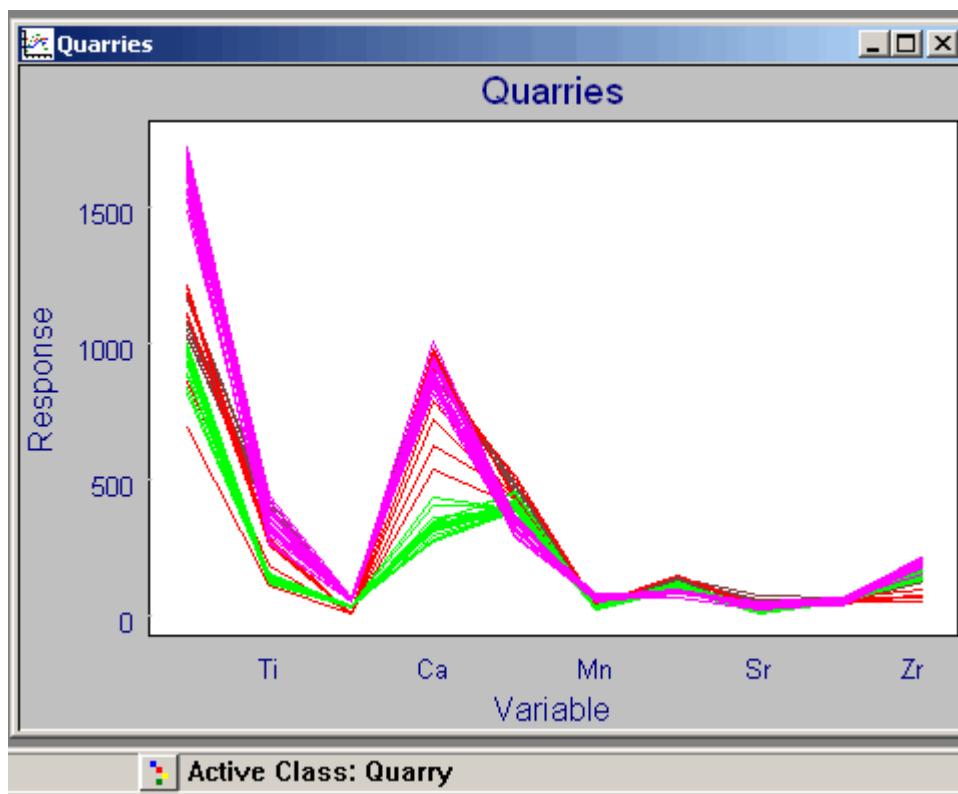
All sample oriented scatter and line plots are thereafter colored by class. Samples with a value of 1 in the active class variable take on the first color in the Color Sequence, samples with a value of 2 have the second color in the sequence, *etc.* Again, when the number of categories or the category value exceeds the number of colors in the sequence, the color sequence wraps. Thus, a class value of nine assumes the third color of a six color sequence.

Why use category-based color mapping? Samples having the same category value are similar somehow. The similarity should be discernible in either the raw data, in classification algorithm results or in a sample dendrogram. Therefore, when we look at line plots of many samples of different categories, we expect traces of properly classified samples to group together as a function of category. Color mapping to class facilitates the verification of that expectation. Similarly, scatter plots of categorized samples should show clustering by color (*i.e.*, by category): data points of a cluster should plot near to one another, a fact easily confirmed when the points are colored. The presence of clusters containing more than one color suggests either problems with the assigned categories or that the axes (variables) chosen for that scatter plot view do not distinguish the different categories.

Note: *When no class variable is active, color mapping is simply tied to the table view row or column index. Result objects which are not sample oriented, for example, the Factor Select object, are unaffected by activating a class variable because color mapping to class is tied only to a sample's class value.*

After activating a class as described above, the name of the class appears at the bottom of the screen as shown below.

Figure 12.44
Status bar showing
the active class



To deactivate a class variable, so that the color mapping reverts to the serial order number of the samples in the spreadsheet,

- Select No Class from the Edit menu

or

- Click the Active Class button in the Status Bar to display the Active Class selection dialog.
- Choose None and click OK

Note: *If a data set has only one class variable, that variable is activated automatically when the file is loaded. If more than one class variable, none is activated on file load.*

Tables

Contents

Introduction to the Spreadsheet	13-1
Navigating the Spreadsheet	13-2
Selecting Data	13-5
Editing Data	13-7
Class Variables	13-19
Creating Subsets from Tables	13-20

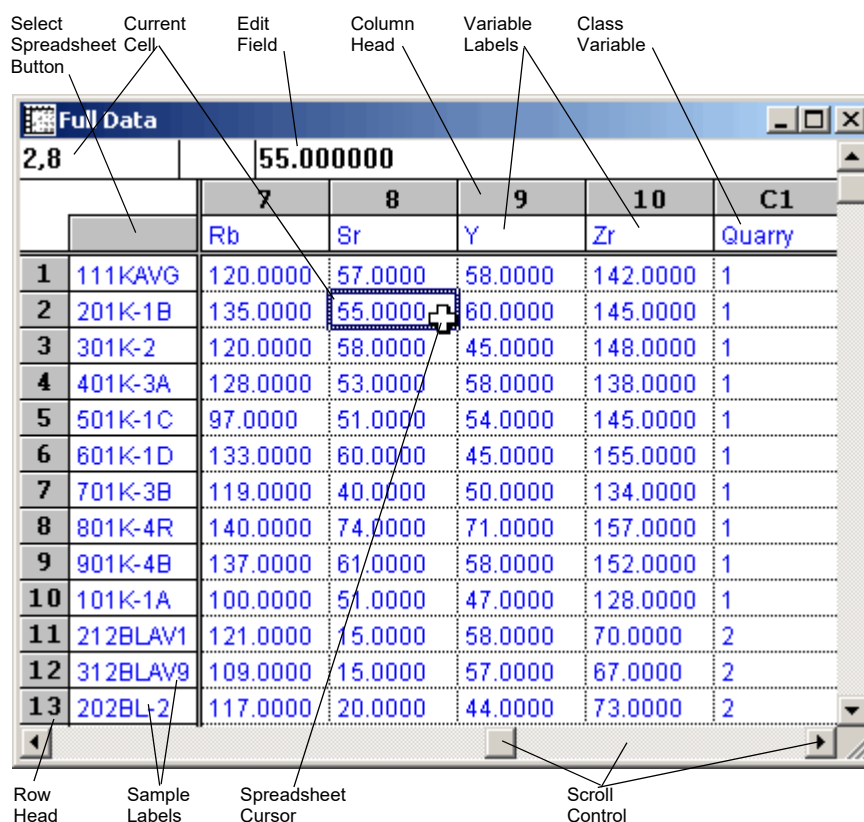
This chapter introduces the Pirouette spreadsheet and discusses how to move around in it, how to enter and modify data, how to specify variable types and fill in missing values. Also included are explanations of how to create/modify subsets and activate class variables from the spreadsheet.

Introduction to the Spreadsheet

In [Figure 13.1](#) the various features of a Pirouette spreadsheet are identified. In appearance it resembles other spreadsheets, being composed of columns and rows of cells which contain data values. However, Pirouette was not designed for the business functions most spreadsheets perform. Instead, it was created to facilitate processing of large multivariate data sets. Although the data portion of the spreadsheet operates much like other spreadsheets, the name fields for columns and rows have behaviors unique to Pirouette. Note also that the data region of a Pirouette spreadsheet is divided into three parts, holding data from independent, dependent, and categorical variables.

Most algorithm results can be displayed in a tabular view. In this case, however, some actions appropriate for raw data are disabled. Even though spreadsheet views of raw data and algorithm results have some differences, we will nevertheless refer to both as tables.

Figure 13.1
A Pirouette spreadsheet or table



Navigating the Spreadsheet

There are a variety of ways to move around a Pirouette spreadsheet. Some methods move the active cell while others merely move your view to a different position in the spreadsheet.

MOVING THE ACTIVE CELL

The active cell, which has a thick border, is where editing occurs (see “Editing Data” on page 13-7). By clicking on a new cell with the mouse or by one of the following methods, you can change the active cell.

Keyboard Control

The active cell is most easily moved with the arrow keys on the keyboard. In addition, Tab and Shift-Tab move the cell right and left, respectively, and Enter and Shift-Enter move the cell down and up, respectively. However, when a range of cells has been selected, these behaviors change. Then, the active cell can be moved within a selected range only with the Tab and Enter key combinations. In addition, movement with these keys will “wrap” within the selected cell range. For example, pressing Tab with the active cell in the last selected cell in a row moves to the first selected cell in the next row. If you use an arrow key when a range has been selected, the active cell moves and the range is deselected.

Shortcuts for moving the active cell to a different position in the spreadsheet are summarized in the following table.

Table 13.1
Keyboard shortcuts
for moving active
cell

Key(s)	Action
→	One cell to right
↓	One cell down
←	One cell to left
↑	One cell up
Ctrl+→	To last cell in row
Ctrl+↓	To last cell in column
Ctrl+←	To first cell in row
Ctrl+↑	To first cell in column
Tab	One cell to right
Shift+Tab	One cell to left
Enter	One cell down
Shift+Enter	One cell up
Home	To first cell in row
End	To last cell in column
Ctrl+Home	To first cell
Ctrl+End	To last cell
Page Down	One page down
Page Up	One page up
Ctrl+Page Down	One page to right
Ctrl+Page Up	One page to left

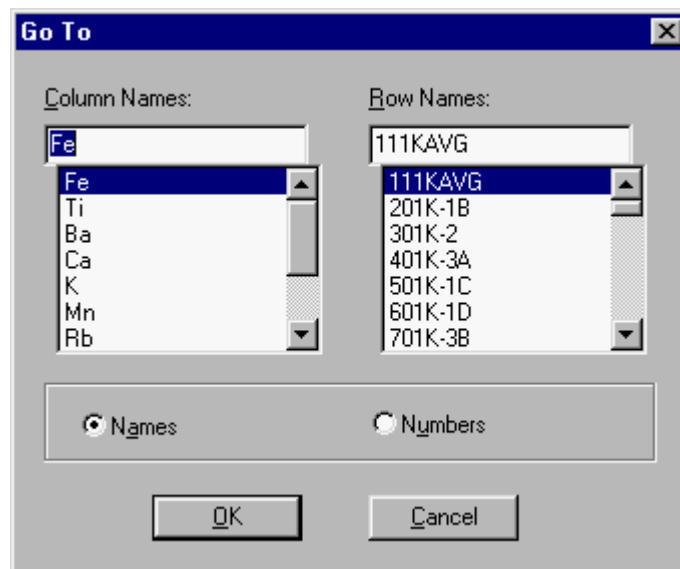
Go To

If your data set is very large, you may want to move the active cell directly to a specific location. The Go To menu item provides this capability.

- Select Go To from the Edit menu

and the following dialog box will be presented.

Figure 13.2
Go To dialog box



Select a column and/or a row name by clicking on a name in each list. You can select the destination cell by number or name by clicking on the appropriate Selection Type radio button. When you click on OK, your view of the table moves to that region and the cell specified becomes the active cell.

Note: *If columns or rows are highlighted before using Go To, they will become unhighlighted afterwards. If you don't want to lose the highlight status, you should navigate to your destination with a means other than the Go To function.*

MOVING TO A NEW PAGE

Often you may need to jump to different views of the data without moving the active cell.

Scroll Bars

Because Pirouette is designed to handle extremely large data sets, scroll bars are incorporated in all tabular view windows. Scroll bars are shaded regions to the right and bottom of spreadsheet windows which contain scroll arrows and elevator boxes. The following descriptions are a brief summary of their functions; for complete information, refer to your Windows manual.

Scroll Arrows

Each click on a scroll arrow shifts the view in the spreadsheet by one row or column. Continuously pressing a scroll arrow causes continuous scrolling.

Elevator Box

The elevator box, sometimes called the thumb, is the small square located within the scroll bar region. Click–drag the elevator box to rapidly jump from one spreadsheet region to another. To move the view to the last page of data, drag the elevator box to the end of the scroll bar.

Page Scroll Region

Click in the shaded region below/above (or left/right of) the elevator box to scroll the display by a screenful.

Variable Type Blocks

In Pirouette, the different types of variables occupy separate spreadsheet blocks. To navigate between these different blocks, use the variable block buttons in the Pirouette ribbon. Clicking on the X, C and Y buttons displays at the left edge of table view, respectively, the first column of the X block (independent), Y block (dependent), and C block (categorical) variables. In the following figure, the first view has the active cell in the X block; the second view results from clicking on the C button. If there are no variables of a block type, the corresponding button is disabled.

Figure 13.3
Moving in the spreadsheet with the C-block button

689.000000				
	1	2	3	
	Fe	Ti	Ba	
1	111KAVG	1100.0000	390.0000	55.0000
2	201K-1B	1173.0000	417.0000	54.0000
3	301K-2	1164.0000	404.0000	56.0000
4	401K-3A	1030.0000	373.0000	59.0000
5	501K-1C	1077.0000	373.0000	55.0000
6	601K-1D	1080.0000	403.0000	53.0000
7	701K-3B	1020.0000	360.0000	59.0000
8	801K-4R	1050.0000	396.0000	56.0000
9	901K-4B	1100.0000	373.0000	53.0000
10	101K-1A	1069.0000	375.0000	51.0000
11	212BLAV1	863.0000	183.0000	8.0000
12	312BLAV9	1108.0000	289.0000	7.0000
13	202BL-2	1210.0000	276.0000	10.0000

1.11			
	C1	C2	13
	Quarry	Cat 2	
1	111KAVG	1	3
2	201K-1B	1	3
3	301K-2	1	3
4	401K-3A	1	3
5	501K-1C	1	3
6	601K-1D	1	3
7	701K-3B	1	3
8	801K-4R	1	3
9	901K-4B	1	3
10	101K-1A	1	3
11	212BLAV1	2	2
12	312BLAV9	2	2
13	202BL-2	2	2

Selecting Data

Many actions within Pirouette depend on the selection (highlighting) of either rows and/or columns of data. Often entire rows or columns are selected, as during Insert and Delete (described below). Other actions, such as Copy, may require that a single cell or a small range of cells be selected but also work on entire columns or rows. As in other spreadsheets, to select more than a single cell, you click in a cell, then drag to another before releasing the mouse button. All cells bracketed by the drag area are selected following this action. The following figure shows the result of dragging over a small range of cells.

Figure 13.4
Highlighting a range
of cells

3,2		404.000000			
		1	2	3	4
		Fe	Ti	Ba	Ca
1	111KAVG	1100.000000	390.000000	55.000000	920.000000
2	201K-1B	1173.000000	417.000000	54.000000	961.000000
3	301K-2	1164.000000	404.000000	56.000000	916.000000
4	401K-3A	1030.000000	373.000000	59.000000	920.000000
5	501K-1C	1077.000000	373.000000	55.000000	888.000000
6	601K-1D	1080.000000	403.000000	53.000000	919.000000
7	701K-3B	1020.000000	360.000000	59.000000	883.000000
8	801K-4R	1050.000000	396.000000	56.000000	924.000000
9	901K-4B	1100.000000	373.000000	53.000000	910.000000
10	101K-1A	1069.000000	375.000000	51.000000	958.000000
11	212BLAV1	863.000000	183.000000	8.000000	626.000000
12	312BLAV9	1108.000000	289.000000	7.000000	783.000000

To select an entire row or column, click on the row or column index. This highlights the entire row/column, including the name. Similarly, to select a range of rows or columns, click on one index, drag the cursor to another then release. All rows/columns included from the initial click to the release point become highlighted. To select ranges that are discontinuous or too far apart to click-drag, use Shift-clicking to extend and Ctrl-clicking to append a selection. These techniques are described in “[Selecting in Lists and Tables](#)” on page 10-1.

Because of Pirouette’s dynamic linking among objects, when you select samples (or variables) in a data table, that highlighting appears in all other objects containing the same samples (or variables). Linking is discussed in “[Charts](#)” on page 12-1.

To the left of the first variable name and above the first sample name is a shaded “cell” that has no content; it is the Select Table button which highlights the entire spreadsheet. This shortcut is handy for copying the entire table contents to another application.

Figure 13.5
The Select Table
button

1,1		1100.000000		
		1	2	3
+		Fe	Ti	Ba
1	111KAVG	1100.000000	390.000000	55.000000
2	201K-1B	1173.000000	417.000000	54.000000
3	301K-2	1164.000000	404.000000	56.000000
4	401K-3A	1030.000000	373.000000	59.000000
5	501K-1C	1077.000000	373.000000	55.000000

Note: To select all samples, without selecting variable names, select the first sample, then hold down Ctrl+Shift and hit the down arrow.

Similar to the shortcuts for moving the active cell, there are shortcuts for selecting entire columns and rows. These are summarized in the following table. In the first column, go to the indicated region by a mouse click or by one of the shortcuts in Table 13.1 or in this table.

Table 13.2
Keyboard shortcuts
for selecting
columns and rows

Do First	Keys	Action
Go to first cell in row	Shift+Ctrl+→	Selects entire row
Go to first cell in column	Shift+Ctrl+↓	Selects entire column
Select an entire row	Shift+↓	Adds next row to selection
Select an entire row	Shift+↑	Adds previous row to selection
Select an entire row	Shift+Ctrl+↓	Adds all following rows to selection
Select an entire row	Shift+Ctrl+↑	Adds all preceding rows to selection
Select an entire column	Shift+→	Adds next column to selection
Select an entire column	Shift+←	Adds previous column to selection
Select an entire column	Shift+Ctrl+→	Adds all following columns to selection
Select an entire column	Shift+Ctrl+←	Adds all preceding columns to selection

Editing Data

If you are familiar with Microsoft Excel, then you already know how to use Pirouette's spreadsheet. For the spreadsheet novice, the following discussion reviews basic editing actions.

Note that all of these functions work with raw data subsets in Pirouette. However, algorithm results are not modifiable, so some editing functions in table views of algorithm results are disabled. Nevertheless, you can always copy results to other destinations.

Note: *If you have already run an algorithm on a subset, changing a raw data value or a variable type may invalidate the results: if algorithm results exist which were computed from the data, they will no longer relate to the modified raw data. Thus, these invalidated results, and any corresponding charts, will be discarded (after a warning).*

However, if the changes are made to excluded data, then the results for that exclusion set will NOT be thrown out (new in version 4.5)

CHANGING DATA VALUES

To change a value in the active cell, simply type the new value. When you finish entering the value, accept the change either by clicking in another cell or using one of the keyboard methods of moving the active cell.

To enter a value into a cell not currently part of the data range (*i.e.*, the column or row where you want the value to go is currently completely blank), type the value into the cell, as above. When you accept the entry, all other cells in that row or column are filled with the missing value character (“*”) so that a rectangular data area is maintained. For example, in the following figure, a value was added to the right of an existing data table, and the remainder of the column was filled with missing values.

Figure 13.6
Entering a value into
an empty cell

Full Data		7	8	9	10	11	12
		Rb	Sr	Y	Zr	var11	
1	111KAVG	120.00000	57.000000	58.000000	142.00000	15.500000	
2	201K-1B	135.00000	55.000000	60.000000	145.00000	*	
3	301K-2	120.00000	58.000000	45.000000	148.00000	*	
4	401K-3A	128.00000	53.000000	58.000000	138.00000	*	
5	501K-1C	97.000000	51.000000	54.000000	145.00000	*	
6	601K-1D	133.00000	60.000000	45.000000	155.00000	*	
7	701K-3B	119.00000	40.000000	50.000000	134.00000	*	
8	801K-4R	140.00000	74.000000	71.000000	157.00000	*	
9	901K-4B	137.00000	61.000000	58.000000	152.00000	*	
10	101K-1A	100.00000	51.000000	47.000000	128.00000	*	
11	212BLAV1	121.00000	15.000000	58.000000	70.000000	*	
12	312BLAV9	109.00000	15.000000	57.000000	67.000000	*	

You will need to supply data for all missing values before processing with any of the algorithms.

MANIPULATING RANGES OF DATA

You work with ranges of data when you move blocks of data from one table region to another.

Cut, Copy, Paste, and Clear

Blocks of data within Pirouette are moved via the Clipboard, a memory buffer which stores large amounts of data temporarily. Data are put into the clipboard with the Cut or Copy commands and are retrieved with the Paste command. To use any of these commands, you first select a range of data, then perform the appropriate action.

- Select Cut from the Edit menu

and the contents of data in the selected range are placed into the clipboard. They are removed from the table when a Paste next occurs. To remove data immediately, without affecting the contents of the clipboard, choose Clear and missing value characters will appear in the selected range.

To duplicate data to other parts of a data table,

- Select Copy from the Edit menu

and the data in the selected range are placed on the clipboard. When you have highlighted a destination range,

- Select Paste from the Edit menu

The Paste command overwrites the contents of the destination cells.

If you copy a range of more than one cell and select a single destination cell, Pirouette assumes that the destination matches the origin range and pastes the entire clipboard contents accordingly. If, however, you select more than a single destination cell and that shape does not match the copied range of cells, an error message signals the inconsistency and the paste fails. On the other hand, if the destination shape is a “multiple” of the origin shape, this is considered a match and the clipboard contents are pasted multiple times.

Note: *To repeat a single value, as when assigning values to a class variable where many samples have the same category value, type in the first cell value, copy that cell, select the range of cells to contain the same value and paste.*

Because Pirouette shares the clipboard with other Windows applications, you can exchange data with other programs. For example, you can paste data copied from Excel into Pirouette. Similarly, to include Pirouette results in a report you are writing, copy them from Pirouette and paste them into your word processor.

Insert and Delete

To create space for new data in your table,

- Highlight a row or column
- Select Insert from the Edit menu

The new row or column is created above the row or to the left of the column selected. If you select contiguous rows or columns and insert, that number of rows or columns is inserted. Inserted rows or columns will be of the same type as the first row or column selected.

Note: *You are not allowed to do an insert if the selection range includes variables of more than one type.*

Highlighted rows and columns can be deleted via the Delete item on the Edit menu.

CHANGING VARIABLE TYPES

To change a variable type,

- Highlight one or more contiguous columns of the same type
- Select Column Type from the Edit menu
- Choose the desired variable type from the submenu

When a column's type changes, the column moves to the end of the appropriate spreadsheet block and the column index changes to reflect the new column type. The next figure shows the changes which occur when an independent variable is converted to a class variable. The variable named Rb moves to the area of the spreadsheet containing categorical variables and the column index is preceded by a C, indicating its class variable status.

Figure 13.7
Changing a variable
type

The figure consists of two screenshots of the 'Full Data' window. The top screenshot shows the initial state where column 7 is 'Rb' (independent variable) and column 12 is 'Quarry' (class variable). The bottom screenshot shows the result after converting 'Rb' to a class variable: 'Rb' has moved to column C2 and its values are truncated to integers, while 'Quarry' has moved to column C1.

Full Data		7	8	9	10	C1	12
1,7		120.000000					
		Rb +	Sr	Y	Zr	Quarry	
1	111KAVG	120.000000	57.000000	58.000000	142.000000	1	
2	201K-1B	135.000000	55.000000	60.000000	145.000000	1	
3	301K-2	120.000000	58.000000	45.000000	148.000000	1	
4	401K-3A	128.000000	53.000000	58.000000	138.000000	1	
5	501K-1C	97.000000	51.000000	54.000000	145.000000	1	
6	601K-1D	133.000000	60.000000	45.000000	155.000000	1	

Full Data		7	8	9	C1	C2	12
1,7		57.000000					
		Sr	Y	Zr	Quarry	Rb	
1	111KAVG	57.000000	58.000000	142.000000	1	120	
2	201K-1B	55.000000	60.000000	145.000000	1	135	
3	301K-2	58.000000	45.000000	148.000000	1	120	
4	401K-3A	53.000000	58.000000	138.000000	1	128	
5	501K-1C	51.000000	54.000000	145.000000	1	97	
6	601K-1D	60.000000	45.000000	155.000000	1	133	

In Pirouette, categorical variables are integers. Therefore, if you convert an independent or dependent column variable to a class variable, a warning that values will be truncated is presented.

Note: The truncation that occurs upon converting an independent or dependent variable column to a category type of variable is not reversible.

SORTING DATA

If the sequence of data collection or presentation is not critical, it may be desirable to sort data into an order that is more compelling. With the Pirouette Sort feature, you can rearrange your data as a function of sorting “keys” in either the sample or variable domain.

- If rows are selected, the sort key is either the variable containing the active cell or the sample name, and selected rows will be sorted.
- If columns are selected, the sort key is either the sample containing the active cell or the variable name, and selected columns will be sorted.

Consider the following example. To sort the columns in alphabetical order by element,

Figure 13.8
The ARCH data set
with 10 columns
highlighted

Full Data		1100.000000					
		1	2	3	4	5	6
		Fe	Ti	Ba	Ca	K	Mn
1	111KAVG	1100.00	390.00	55.00	920.00	460.00	45.00
2	201K-1B	1173.00	417.00	54.00	961.00	441.00	47.00
3	301K-2	1164.00	404.00	56.00	916.00	446.00	42.00
4	401K-3A	1030.00	373.00	59.00	920.00	487.00	38.00
5	501K-1C	1077.00	373.00	55.00	888.00	455.00	38.00
6	601K-1D	1080.00	403.00	53.00	919.00	442.00	41.00
7	701K-3B	1020.00	360.00	59.00	883.00	473.00	43.00
8	801K-4R	1050.00	396.00	56.00	924.00	482.00	48.00
9	901K-4B	1100.00	373.00	53.00	910.00	477.00	51.00
10	101K-1A	1069.00	375.00	51.00	958.00	429.00	42.00
11	212BLAV1	863.00	183.00	8.00	626.00	452.00	34.00
12	312BLAV9	1108.00	289.00	7.00	783.00	426.00	41.00

- Highlight columns 1 through 10 as shown above
- Select Sort from the Edit menu to open the Sort dialog box shown below

Figure 13.9
Sort dialog box

Sort

Type

Column Row

Column Names Row Names

Order

Ascending Descending

OK Cancel

Columns were selected which disables row sorting choices. To sort by column names in alphabetical order:

- Select Column Names
- Choose Ascending
- Click on OK

The selected columns of data are then rearranged as shown below

Figure 13.10
The ARCH data with
columns sorted
alphabetically

Full Data		55.000000					
1,1		1	2	3	4	5	6
		Ba	Ca	Fe	K	Mn	Sr
1	111KAVG	55.00	920.00	1100.00	460.00	45.00	57.00
2	201K-1B	54.00	961.00	1173.00	441.00	47.00	55.00
3	301K-2	56.00	916.00	1164.00	446.00	42.00	58.00
4	401K-3A	59.00	920.00	1030.00	487.00	38.00	53.00
5	501K-1C	55.00	888.00	1077.00	455.00	38.00	51.00
6	601K-1D	53.00	919.00	1080.00	442.00	41.00	60.00
7	701K-3B	59.00	883.00	1020.00	473.00	43.00	40.00
8	801K-4R	56.00	924.00	1050.00	482.00	48.00	74.00
9	901K-4B	53.00	910.00	1100.00	477.00	51.00	61.00
10	101K-1A	51.00	958.00	1069.00	429.00	42.00	51.00
11	212BLAV1	8.00	626.00	863.00	452.00	34.00	15.00
12	312BLAV9	7.00	783.00	1108.00	426.00	41.00	15.00

If you use the Select Table button (see “Selecting Data”) to highlight all samples and variables simultaneously, the Sort dialog box has all options enabled. In this situation, the sort key is determined by type you choose, *e.g.*, row name or column.

Similarly, if you have at least 2 rows and 2 columns selected at the same time, all sort type options are available.

TRANSPOSE

Some data sources store their sample values in columns rather than in rows as is done in Pirouette. This may be required when using a spreadsheet such as Microsoft Excel (before 2007) which has a limit of 256 columns but your data have many more variables. If it is possible to modify the structure of the data in the source file to match that in the Pirouette ASCII format, it is possible to read these data into Pirouette as their transpose (see Table 14.8, “Transposed ASCII with row labels,” on page 14-8).

Alternatively, Pirouette offers a Transpose function that will swap columns for rows.

- File > Transpose

Note that any subsets and computed results already prepared will be lost by this action (a message will warn you beforehand). Also, Pirouette will preserve the class and variable name prefixes (see page 14-8) in the event that you intend to revert to the original configuration with another transpose action.

FINDING MISSING VALUES

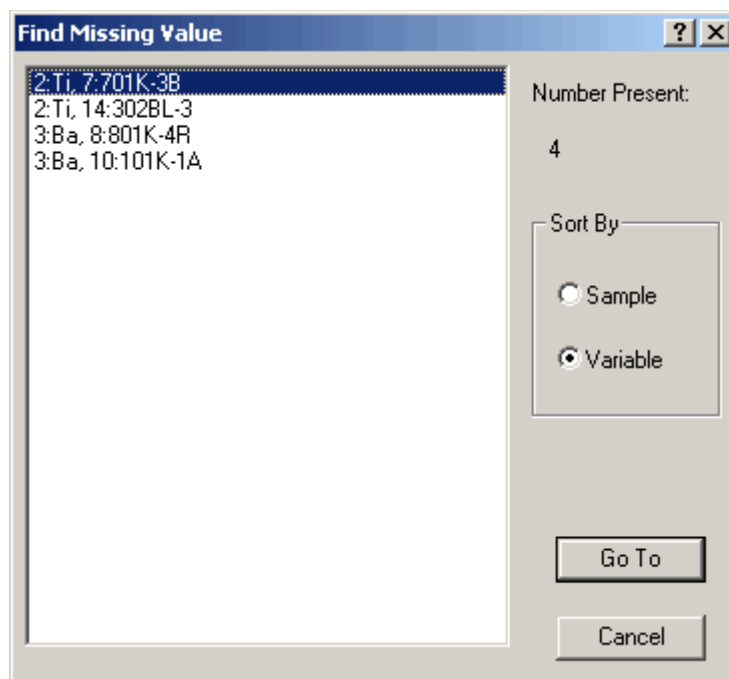
Not every source of data generates completely rectangular data matrices, which you may notice after merging different data sets into Pirouette. In addition, you may encounter situations in which not all measurements can be made for every sample. When such data are entered into Pirouette's spreadsheet, the missing values show up as asterisks.

For small data sets, it is easy to spot the missing values by simply looking at the table view of your data. However, if the data set is large so that scrolling around in the table to find the missing values would be tedious, you can locate them with a special function. With the table view active,

- Edit > Find Missing Values ...

which opens the following dialog box.

Figure 13.11
Finding Missing
Values dialog



The first 1000 missing values found will be listed, and the number of missing values present will be shown at the top of the dialog. Each item in the list is labeled by its coordinates in the table, in the following syntax:

Variable number:Variable name, Sample number:Sample name

You can sort this list in either variable or sample order. For example, it will be easier to find blocks of samples missing values in the same variable if grouped by variable. By clicking on an item in the list then on the Go To button, the table will be displayed with the selected cell in the upper left position in the window.

FILLING MISSING VALUES

Pirouette's multivariate algorithms will not run on sets containing missing values. Several ways to fill missing values are provided, including substituting the missing value with zeroes, with a number dependent on neighboring values, or with prescribed values.

The middle case involves three choices: filling with a vector's mean, a vector's median or interpolating from adjacent values.

Note: Fill operations ignore the exclusion state of columns and rows.

Mean, Median, Interpolated Fill

If samples (rows) are selected, the mean or median is computed across the row containing the missing value (X values only), and interpolated values are derived from row neighbors. When variables (columns) are selected, the analogous column fill occurs.

For example, in the data set shown below, there is one missing value.

Figure 13.12
An example of filling missing data

		1	2	3	4	5
		Fe	Ti	Ba	Ca	K
1	212BLAV1	863.0000	183.0000	8.0000	626.0000	452.0000
2	312BLAV9	1108.0000	289.0000	7.0000	783.0000	426.0000
3	202BL-2	1210.0000	276.0000	10.0000	966.0000	430.0000
4	302BL-3	1205.0000	291.0000	10.0000	975.0000	420.0000
5	502BL-6	1100.0000	267.0000	*	910.0000	500.0000
6	602BL-7	1100.0000	280.0000	10.0000	872.0000	515.0000
7	112BLAV7	689.0000	114.0000	9.0000	534.0000	404.0000
8	102BL-1	1186.0000	257.0000	10.0000	940.0000	431.0000
9	702BL-8	860.0000	182.0000	7.0000	722.0000	418.0000

Highlight either the corresponding row or column (not both as shown in the figure), then

- Select Fill from the Edit menu
- Select either Mean Fill, Median fill or Interpolated Fill from the submenu

The six possible results from these forms of missing-value filling are shown below.

Table 13.3
Filling missing value shown in Figure 13.12

Method	Result
Row-wise Mean Fill	694.25
Row-wise Median Fill	705.00
Row-wise Interpolation Fill	588.50
Column-wise Mean Fill	8.88
Column-wise Median Fill	9.50
Column-wise Interpolation Fill	10.00

The fill function has utility beyond replacing sporadic missing values. In creating new data columns, especially for categorical variables, fill can be used to insert a default value. For example, to create a new class variable column,

- Highlight an existing class variable column
- Select Insert from the Edit menu

A new column appears, in which all cells are initially filled with the missing value character as shown below.

Figure 13.13
Filling a blank
column

Full Data						
1,11		*				
		8	9	10	C1	C2
		Sr	Y	Zr	var11	Quarry
1	111KAVG	57.0000	58.0000	142.0000	*	1
2	201K-1B	55.0000	60.0000	145.0000	*	1
3	301K-2	58.0000	45.0000	148.0000	*	1
4	401K-3A	53.0000	58.0000	138.0000	*	1
5	501K-1C	51.0000	54.0000	145.0000	*	1
6	601K-1D	60.0000	45.0000	155.0000	*	1
7	701K-3B	40.0000	50.0000	134.0000	*	1
8	801K-4R	74.0000	71.0000	157.0000	*	1
9	901K-4B	61.0000	58.0000	152.0000	*	1
10	101K-1A	51.0000	47.0000	128.0000	*	1
11	212BLAV1	15.0000	58.0000	70.0000	*	2
12	312BLAV9	15.0000	57.0000	67.0000	*	2
13	202BL-2	20.0000	44.0000	73.0000	*	2

Notice that the whole column remains selected and the first cell in the column is the active or current cell, indicated by its thicker border.

- Type **1** then press Enter

To place **1** into the class variable for all of the remaining samples in this set,

- Select Fill from the Edit menu
- Choose By Interpolation

and the task is completed, as shown below.

Figure 13.14
Filling a new column
with a constant

Full Data						
1,11		1.000000				
		8	9	10	C1	C2
		Sr	Y	Zr	var11	Quarry
1	111KAVG	57.0000	58.0000	142.0000	1	1
2	201K-1B	55.0000	60.0000	145.0000	1	1
3	301K-2	58.0000	45.0000	148.0000	1	1
4	401K-3A	53.0000	58.0000	138.0000	1	1
5	501K-1C	51.0000	54.0000	145.0000	1	1
6	601K-1D	60.0000	45.0000	155.0000	1	1
7	701K-3B	40.0000	50.0000	134.0000	1	1
8	801K-4R	74.0000	71.0000	157.0000	1	1
9	901K-4B	61.0000	58.0000	152.0000	1	1
10	101K-1A	51.0000	47.0000	128.0000	1	1
11	212BLAV1	15.0000	58.0000	70.0000	1	2
12	312BLAV9	15.0000	57.0000	67.0000	1	2
13	202BL-2	20.0000	44.0000	73.0000	1	2

Fill by Value

On the other hand, you may sometimes wish to fill only a partial range of samples with the same value for which the above procedure would be too tedious. Thus,

- Select the column in which the missing values occur, AND
- Ctrl+Select the rows for which you wish to insert specific values

so that there are both rows and columns selected.

Figure 13.15
Select rows and columns with missing values

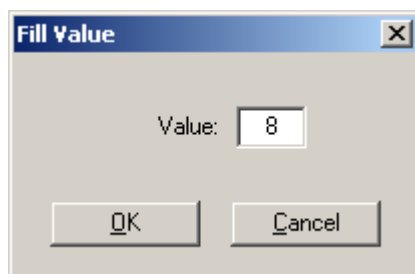
Full Data						
1,11		*				
		8	9	10	C1	C2
		Sr	Y	Zr	var11	Quarry
1	111KAVG	57.0000	58.0000	142.0000	*	1
2	201K-1B	55.0000	60.0000	145.0000	*	1
3	301K-2	58.0000	45.0000	148.0000	*	1
4	401K-3A	53.0000	58.0000	138.0000	*	1
5	501K-1C	51.0000	54.0000	145.0000	*	1
6	601K-1D	60.0000	45.0000	155.0000	*	1
7	701K-3B	40.0000	50.0000	134.0000	*	1
8	801K-4R	74.0000	71.0000	157.0000	*	1
9	901K-4B	61.0000	58.0000	152.0000	*	1
10	101K-1A	51.0000	47.0000	128.0000	*	1
11	212BLAV1	15.0000	58.0000	70.0000	*	2
12	312BLAV9	15.0000	57.0000	67.0000	*	2
13	202BL-2	20.0000	44.0000	73.0000	*	2

When these selections have been prepared,

- Select Fill from the Edit menu
- Choose With Value from the submenu

and a dialog box is presented requesting a value to be used as the fill value:

Figure 13.16
Fill with a value dialog box



Note that only the missing values that fall on the intersections of selected columns and rows will be filled with the Fill With Value approach.

Figure 13.17
Cells filled by value

Full Data						
1,11		8.000000				
		8	9	10	C1	C2
		Sr	Y	Zr	var11	Quary
1	111KAVG	57.0000	58.0000	142.0000	8	1
2	201K-1B	55.0000	60.0000	145.0000	8	1
3	301K-2	58.0000	45.0000	148.0000	8	1
4	401K-3A	53.0000	58.0000	138.0000	8	1
5	501K-1C	51.0000	54.0000	145.0000	8	1
6	601K-1D	60.0000	45.0000	155.0000	8	1
7	701K-3B	40.0000	50.0000	134.0000	8	1
8	801K-4R	74.0000	71.0000	157.0000	8	1
9	901K-4B	61.0000	58.0000	152.0000	8	1
10	101K-1A	51.0000	47.0000	128.0000	8	1
11	212BLAV1	15.0000	58.0000	70.0000	*	2
12	312BLAV9	15.0000	57.0000	67.0000	*	2
13	202BL-2	20.0000	44.0000	73.0000	*	2

Fill as Mask

A specialized fill is also available for use with certain transforms which use a mask row (see “Using a Mask” on page 4-11). This usually requires that you designate a row in your data set which you do not intend to be processed; it is usually excluded in the subsets you will use. Either insert a new row to be used as the mask or clear its contents entirely.

- Highlight the row to be used as the mask
- Ctrl+Select the columns for which the mask will be active

You can select the columns from the spreadsheet or by using the range tool in a line plot of the data.

Figure 13.18
Select columns of a
Mask row

Full Data							
1,5		*					
		1	2	3	4	5	6
		Fe	Ti	Ba	Ca	K	Mn
1	smp1	*	*	*	*	*	*
2	111KAVG	1100.0000	390.0000	55.0000	920.0000	460.0000	45.0000
3	201K-1B	1173.0000	417.0000	54.0000	961.0000	441.0000	47.0000
4	301K-2	1164.0000	404.0000	56.0000	916.0000	446.0000	42.0000
5	401K-3A	1030.0000	373.0000	59.0000	920.0000	487.0000	38.0000
6	501K-1C	1077.0000	373.0000	55.0000	888.0000	455.0000	38.0000
7	601K-1D	1080.0000	403.0000	53.0000	919.0000	442.0000	41.0000
8	701K-3B	1020.0000	360.0000	59.0000	883.0000	473.0000	43.0000
9	801K-4R	1050.0000	396.0000	56.0000	924.0000	482.0000	48.0000
10	901K-4B	1100.0000	373.0000	53.0000	910.0000	477.0000	51.0000
11	101K-1A	1069.0000	375.0000	51.0000	958.0000	429.0000	42.0000
12	212BLAV1	863.0000	183.0000	8.0000	626.0000	452.0000	34.0000
13	312BLAV9	1108.0000	289.0000	7.0000	783.0000	426.0000	41.0000

Then,

- Select Fill from the Edit menu
- Choose As Mask from the submenu

All values in the mask row for columns which were highlighted will be filled with ones and all remaining missing values in the row will be filled with zeros.

Figure 13.19
Values filled by Mask

Full Data		1.000000					
1,5		1	2	3	4	5	6
		Fe	Ti	Ba	Ca	K	Mn
1	smp1	0.0000	1.0000	1.0000	0.0000	1.0000	0.0000
2	111KAVG	1100.0000	390.0000	55.0000	920.0000	460.0000	45.0000
3	201K-1B	1173.0000	417.0000	54.0000	961.0000	441.0000	47.0000
4	301K-2	1164.0000	404.0000	56.0000	916.0000	446.0000	42.0000
5	401K-3A	1030.0000	373.0000	59.0000	920.0000	487.0000	38.0000
6	501K-1C	1077.0000	373.0000	55.0000	888.0000	455.0000	38.0000
7	601K-1D	1080.0000	403.0000	53.0000	919.0000	442.0000	41.0000
8	701K-3B	1020.0000	360.0000	59.0000	883.0000	473.0000	43.0000
9	801K-4R	1050.0000	396.0000	56.0000	924.0000	482.0000	48.0000
10	901K-4B	1100.0000	373.0000	53.0000	910.0000	477.0000	51.0000
11	101K-1A	1069.0000	375.0000	51.0000	958.0000	429.0000	42.0000
12	212BLAV1	863.0000	183.0000	8.0000	626.0000	452.0000	34.0000
13	312BLAV9	1108.0000	289.0000	7.0000	783.0000	426.0000	41.0000

PCA Fill

Finally, one method invokes PCA, a familiar algorithm that estimates fill values based on surrounding data¹. In this case, PCA is run on the submatrix composed of the **inter-section** of the selected columns and rows, which must contain at least one missing value. The missing values are replaced with the column means before the algorithm is run. These values are then updated from the reconstructed submatrix via the usual PCA approach. The PCA decomposition is repeated and the missing values reconstructed until convergence to a stable solution is achieved or the maximum number of iterations is exceeded. Because the approach is based on the surrounding data, discontinuous row or column selections can produce undesirable or non-intuitive results.

Note: *It is not necessary to select all rows or all columns. For example, if three rows are selected, one of which contains the missing value, but no column is selected, you can still run the PCA Fill because Pirouette assumes that you meant to select all columns. Similarly, after selecting only a few columns and no rows, PCA Fill will proceed, assuming all rows have been selected, thus used in the algorithm.*

In either event, there must be at least 5 rows or 5 columns selected (or assumed to be selected) before you can run a PCA fill.

Figure 13.20
 Example of PCA Fill;
 a) original data, b)
 data with missing
 value and small
 number of selected
 columns, c) data with
 filled value

a		1	2	3	4	5
		900	920	940	960	980
1	S001	0.2146	0.2364	0.2110	0.1979	0.1971
2	S002	0.2198	0.2402	0.2163	0.2036	0.2028
3	S003	0.2164	0.2379	0.2129	0.2000	0.1992
4	S004	0.2201	0.2403	0.2155	0.2037	0.2031
b		1	2	3	4	5
		900	920	940	960	980
1	S001	0.2146	0.2364	0.2110	0.1979	0.1971
2	S002	0.2198	0.2402	0.2163	0.2036	0.2028
3	S003	0.2164	0.2379	0.2129	*	0.1992
4	S004	0.2201	0.2403	0.2155	0.2037	0.2031
c		1	2	3	4	5
		900	920	940	960	980
1	S001	0.2146	0.2364	0.2110	0.1979	0.1971
2	S002	0.2198	0.2402	0.2163	0.2036	0.2028
3	S003	0.2164	0.2379	0.2129	0.2005	0.1992
4	S004	0.2201	0.2403	0.2155	0.2037	0.2031

Class Variables

Class variables in Pirouette provide a basis for color mapping (see “Color Sequence” on page 10-18) and indicate known category assignments when running a classification algorithm. For some regression algorithms, a class variable can be designated to control cross validation; see “Category validation (using the Active Class)” on page 5-21.

A data set may contain several class variables but only one can be the active class variable. Moreover, it is sometimes desirable to have no class variable active.

ACTIVATING A CLASS VARIABLE

There are three ways to activate a class variable. One is via the HCA dendrogram; see “Creating Class Variables” on page 12-27. Another is accomplished from a table view of the raw data:

- Click on the desired class column index
- Choose Activate Class from the Edit menu (or use the Ctrl-K shortcut)

A third method uses the Active Class button in the status bar (see page 12-35).

See “Color Sequence” on page 10-18 and “Plot Colors” on page 12-34 for more information on how color mapping works when a class variable is active.

USING CLASS VARIABLES IN ALGORITHMS

Classification algorithms (KNN, SIMCA and PLS-DA) require that you specify which category variable will be used during model creation. There are, however, different rules regarding the values allowed in a class variable. These rules are summarized in the following table (only integer values are allowed in a class variable).

Table 13.4
Allowed content of
class variable used
in classification
algorithm

	Allowed	Not allowed
KNN	negative, zero, positive values	missing values
SIMCA	negative, positive values	zero, missing values
PLS-DA	negative, positive values	zero, missing values

During prediction, there is no constraint on the values allowed.

Creating Subsets from Tables

We seldom use all the collected data in the final multivariate investigation. Instead we may exclude some samples because they are of an under-represented type or because they were contaminated or otherwise deemed unfit. Pirouette provides a simple mechanism for manipulating data in the pursuit of the optimal subset.

EXCLUDING DATA

Subsets can be created from either table or graphic views. Here we focus on subset creation from tables. For a discussion of subset creation from other views, see [“Creating Subsets from a Graphic”](#) on page 12-32.

Creating a subset from the spreadsheet is a three step procedure:

- Select (highlight) rows and/or columns that you want to exclude
- Select Create Exclude from the Edit menu
- Rename the subset with the Rename item in the Object menu

As an example, let’s create a subset using the ARCH.XLS data file. This data set contains two groups of data. The first 63 rows contain quarry data, while rows 64 to 75 contain information on artifacts. To put the quarry samples in a separate subset.

- Scroll down to where the quarry samples meet the arrowhead information (artifact samples start at row 64)
- Highlight the samples as shown in the figure below

Figure 13.21
Artifact samples
highlighted

Full Data							
64,1		1050.0000					
		1	2	3	4	5	6
		Fe	Ti	Ba	Ca	K	Mn
61	924ANA-1	1675.0000	325.0000	62.0000	930.0000	350.0000	66.0000
62	024ANA-2	1685.0000	353.0000	59.0000	875.0000	318.0000	60.0000
63	124ANA-2	1600.0000	316.0000	54.0000	915.0000	347.0000	83.0000
64	s2112909	1050.0000	195.0000	46.0000	865.0000	400.0000	36.0000
65	s3111309	1010.0000	357.0000	65.0000	900.0000	455.0000	42.0000
66	s4111313	920.0000	320.0000	60.0000	830.0000	440.0000	33.0000
67	s5116953	1000.0000	194.0000	58.0000	965.0000	460.0000	42.0000
68	s402	920.0000	215.0000	6.0000	650.0000	435.0000	37.0000
69	s4121313	780.0000	140.0000	12.0000	605.0000	415.0000	35.0000
70	s5121	680.0000	105.0000	10.0000	460.0000	400.0000	31.0000
71	s1132909	920.0000	127.0000	39.0000	475.0000	430.0000	34.0000
72	s2132910	900.0000	115.0000	37.0000	310.0000	440.0000	34.0000
73	s3132910	920.0000	138.0000	41.0000	350.0000	450.0000	34.0000
74	s4136953	775.0000	110.0000	35.0000	327.0000	337.0000	34.0000
75	s5136953	935.0000	131.0000	38.0000	360.0000	420.0000	37.0000

- Choose Create Exclude from the Edit menu (or use the Ctrl-E shortcut)

By excluding all artifact samples, we have created a subset called Unnamed1 which includes only the quarry samples.

INCLUDING DATA

When starting from the full data set, all rows and columns are already included. Including becomes necessary when one desires to re-include previously excluded data. Highlighting excluded columns or rows and selecting Create Include reverses their inclusion status, and generates a new subset which reflects these changes.

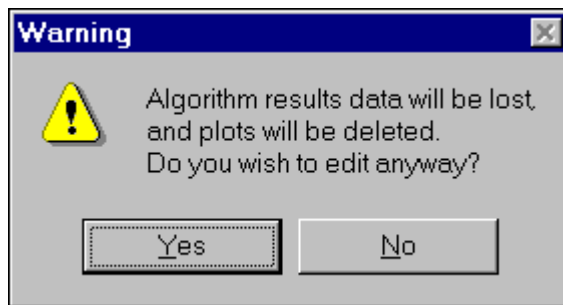
Note: To create an identical but distinct subset, select an already included row or column, then choose Create Include. No change in exclusion status will occur, but a new set will be created.

MODIFYING SUBSETS

To modify an existing subset, without creating a new one, proceed in the same manner as before, but instead of Create Exclude or Create Include, choose Exclude or Include. These actions implement the exclusions and inclusions but do not generate a new entity in the Object Manager. Instead the existing subset is merely updated.

If you modify a subset having algorithm results, those results and associated charts may become invalid. If so, those algorithm results will be discarded after a warning like the one shown below.

Figure 13.22
Warning when a
subset with results is
modified



SAMPLE AND VARIABLE SELECTION

All of the methods above require user interaction to create the subsets. There are also algorithmic methods to accomplish the same goal. “Subsets” on page 11-9 describe procedures for making new subsets of a reduced number of samples or of variables.

References

1. Walczak, B. and Massart, D.L. “Dealing with Missing Data. Part I”. *Chemometrics and Intelligent Laboratory Systems*, (2001), 58:15-27.

Data Input

Contents

Entering New Data	14-1
Opening and Merging Existing Data Files	14-3
Common File Formats	14-5
Other File Formats	14-10

Ask most users of data analysis software what their biggest complaint is, and the answer invariably concerns the difficulties in moving data between different systems and applications. Unfortunately, for the user and the developer, data interchange formats have not yet been standardized. Even now, it is not uncommon to need two or more programs to reformat and convert data from one source such that it can be read by another application. In Pirouette, we try to make all aspects of data entry painless, from pasting into the spreadsheet from other applications to the importing and merging of files from a variety of sources.

Because data entry is such a vital consideration, the different sections of this chapter are illustrated using files included in the Pirouette package. See [“Description of Example Files” in Chapter 9](#) for more information about these files.

There are two ways to prepare data for analysis in Pirouette: by typing data into the spreadsheet or by reading data in from an existing file. Manual data entry is described first; reading of existing data files is covered in the next sections. Finally, Pirouette’s Merge facility is explained.

Entering New Data

For small data files, hand-entering data from within Pirouette is feasible. To generate a blank spreadsheet,

- Select New from the File menu

The cells just to the right of the row indices are for sample names while the cells just below the column indices are for variable names; these name fields can contain text values. All other cells are data cells which can only contain numeric values.

Figure 14.1
A blank Pirouette spreadsheet

		1	2	3	4	5	6
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							

Initially, the first data cell is highlighted with a thicker border, indicating that it is the active cell. If you begin typing, your keystrokes will appear in this cell. Figure 14.2 shows a portion of a spreadsheet to which data has been entered manually. As you enter data, default sample and variable names are generated as needed and a rectangular area is maintained, that is, incomplete data rows or columns are padded with asterisks (*), the missing value indicator. You are permitted to have missing values in a Pirouette spreadsheet, but algorithms will be unable to process data vectors which include missing values. Dealing with missing values is discussed in “Filling Missing Values” on page 13-13.

Figure 14.2
A portion of an spreadsheet in progress

		1	2
1,1		1.45620	
		var1	var2
1	smp1	39.1000	1.4562
2	smp2	76.6000	*
3	smp3	36.3000	*
4	smp4	35.7000	*
5	smp5	31.0000	*
6	smp6	35.7000	*
7	smp7	25.8000	*
8			

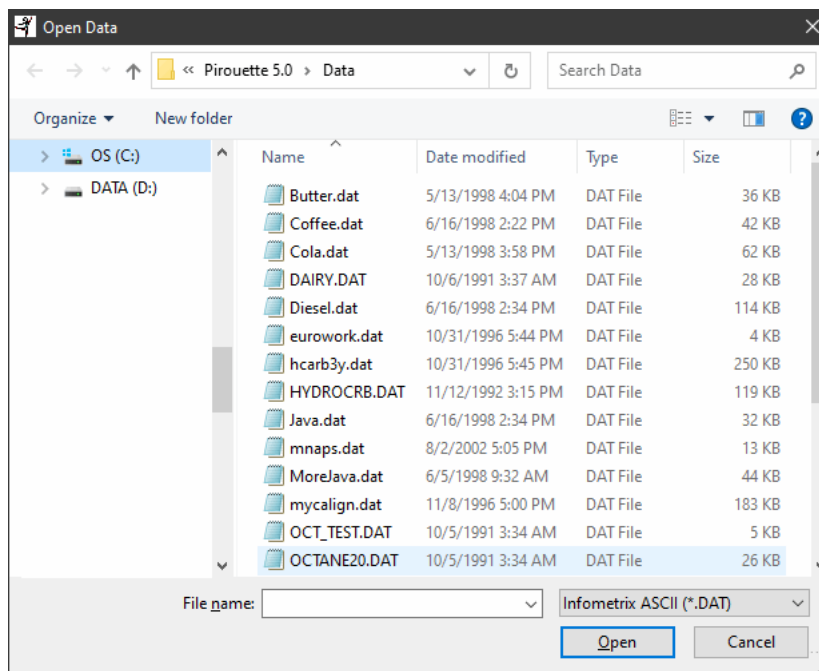
When you have finished hand entering data, you may wish to paste values from other applications into your spreadsheet. This topic is discussed in “Cut, Copy, Paste, and Clear” on page 13-9.

Opening and Merging Existing Data Files

To load data into Pirouette,

- Select Open Data from the File menu.

Figure 14.3
Open Data dialog
box



Using the dialog box shown above, you first navigate the directory structure to find the file you want and modify the File Type filter if necessary. Once the file you seek is displayed, click on its name and click on OK, or, alternatively, double-click on its name. The dialog box will close, and Object Manager will update to show that a file has been loaded. To see a tabular display of data in the file, drag the Full Data set icon to the Pirouette workspace.

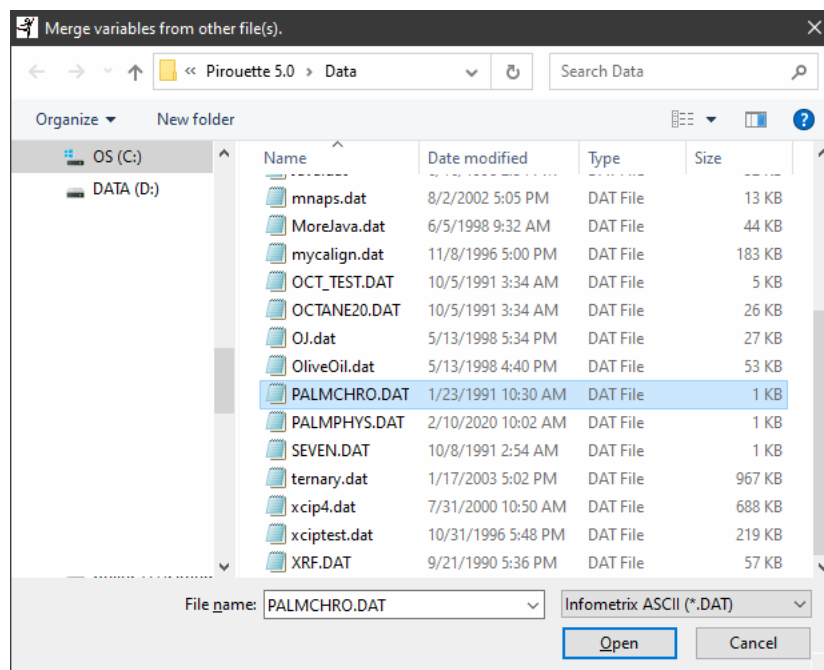
Because you may want to add data with new measurements to an existing file or blend two smaller files into one file, Pirouette includes a facility for merging two or more data files. To demonstrate this feature, we will merge the PALMPHYS.DAT file with another ASCII file called PALMCHRO.DAT. Both files contain data for the same group of samples, but PALMPHYS.DAT contains physical data while PALMCHRO.DAT contains chromatographic data. The new file created by merging these two files will include both the physical and chromatographic variables.

Pirouette allows you to choose whether to merge the data as new samples or new variables by providing two separate menu items. After opening PALMPHYS.DAT (which contains six variables),

- Select the Merge Variables item in the File menu

14 Data Input: Opening and Merging Existing Data Files

Figure 14.4
Merge Variables
dialog box



- Select Infometrix ASCII in the Files of Type pull down list
- Select PALMCHRO.DAT from the DATA subdirectory
- Click on Open

Drag the Full Data item onto the Pirouette workspace; the spreadsheet appears as in the figure below. Notice that the variables in the PALMCHRO . DAT file have been added to the right of the six variables in the PALMPHYS . DAT file, but the columns have been grayed (or excluded).

Figure 14.5
After merging data
file

Full Data		39.099998					
		3	4	5	6	7	8
		var3	var4	var5	var6	Caprilic	Capric
1	smp1	5.10000	188.39999	76.00000	1.40000	0.00000	0.00000
2	smp2	45.10000	203.20000	58.00000	0.40000	0.00000	0.00000
3	smp3	2.50000	198.00000	74.90000	0.40000	0.00000	0.00000
4	smp4	4.70000	200.00000	68.50000	3.30000	0.00000	0.00000
5	smp5	0.00000	196.20000	75.10000	0.00000	0.00000	0.00000
6	smp6	46.70000	197.39999	56.60000	0.40000	0.00000	0.00000
7	smp7	12.00000	194.00000	76.00000	1.60000	5.41000	4.10000

Pirouette treats the merged file as a modification of the old data set. To keep a copy of the original unmerged data, choose the Save Data As option and assign a new name to the merged file; the original file will not be updated.

It is also possible to load a Pirouette file (that is, a file with a .PIR extension), from the Windows Explorer. Simply double-click on the file name or icon, and Pirouette will start, loading the file. If Pirouette is already running, a new instance of Pirouette will be initiated.

Alternatively, you can drag an appropriate data file from the Explorer window into the Pirouette window. This will load the file, replacing the current file which had already been loaded into Pirouette. If you had made changes to the current file, you will be asked if you want to save your changes before proceeding with opening the new file.

You may also drag more than one file into the Pirouette window, and all of the dragged files will be merged into a single data sheet in Pirouette. This is a quick way to start a Pirouette session if your data are originally created in single sample files, often the case with spectroscopic data.

Note: *The order of loading of files following a drag and drop follows certain rules. See “Merging files from different directories” on page 18-2 for tips.*

Common File Formats

Pirouette’s most common file formats are listed in the table below. The DAT and XLS entries have a few specifiers which are discussed in some detail.

Table 14.1
Common Pirouette
file types

Extension	Format Description
.PIR2, .PIR	This is Pirouette’s native format, a binary format for fast loading – this will also store all objects calculated during a Pirouette session. PIR2 - From version 5.0 on this is the default format for Pirouette files. PIR - This format is the legacy form for Pirouette files.
.DAT	An ASCII format, which can be generated by a number of word processors and editors – requires a few formatting specifiers, discussed below
.XLSX, .XLS	XLSX - the standard format created by the Microsoft Excel spreadsheet – requires a few formatting specifics, discussed below. XLS - an older Microsoft Excel format (saved as Excel 97-2003 workbook).

ASCII FILES

An ASCII file must contain appropriate specifiers telling Pirouette which values are numeric data, which are text (*e.g.*, labels), and where the information should be placed in the Pirouette spreadsheet. There are a variety of configurations for the specifiers. Pirouette accepts data alone, data with either sample or variable names and data with both sample and variable names. In addition, you can designate that variables be either of two special types: dependent or class.

In the simplest case, there are no specifiers. All that need be supplied is the raw data as shown in the following table. Samples are oriented as rows and variables as columns (*e.g.*, there are five samples and four variables in the example. Either a space (or several spaces), a comma or a tab separates sample entries, and a carriage return (↵) marks the end of a sample. Generic labels will be created automatically when the ASCII file is read; these can be modified later manually. Pirouette reads samples until it reaches the end of file and assumes that each sample has the same number of variables.

Table 14.2
A simple ASCII file

11	12	13	14	↵
21	22	23	24	↵
31	32	33	34	↵
41	42	43	44	↵
51	52	53	54	↵

If you wish to read an ASCII data set with variable and/or sample labels already in place, the file must contain a set of specifiers followed by the data itself. The specifiers are denoted by a # (pound sign) plus a letter which defines the type of information that follows. The specifiers themselves are not case-sensitive. As before, the entries within a row must be separated by spaces, tabs or commas. Allowed specifiers are:

- #d Dimensionality of the data set. Format is "MxN" where M is the number of variables (*i.e.*, columns) and N is the number of samples (rows), with no spaces between the characters.
- #u Indicates that dimensionality is unknown. Requires that: the word "unspecified" must follow the #u directive; that #r or #p directives must follow; and, that #v directives must be used to indicate variables.
- #c Tells Pirouette that column labels (*i.e.*, variable names) are next
- #r Row labels (*i.e.*, sample names) are next
- #p Same as row labels, except, strip path information from sample names
- #v Tells Pirouette that data for one variable is next; the first "value" is interpreted as the variable name
- #s Data for one sample is next; the first "value" is the sample name

The following nine tables give examples of permutations allowed by the Pirouette ASCII file read. The one you choose will depend on the organization of your data (sample-oriented or variable-oriented files) and whether names are available. In the examples, the values starting with S or V imply that the value is either a sample or variable name.

The simplest ASCII file is a matrix without row or column labels as suggested above. As shown in Table 14.3, the dimension of the data set can also be added. This allows you to load a rectangular matrix in which the data for one sample do not fit on a single line. The previous format without the dimensionality specifier used a carriage return to signal end of line; this format avoids that limitation.

Table 14.3
ASCII file with dimensions but no labels

#d	4x5			
11	12	13	14	
21	22	23	24	
31	32	33	34	
41	42	43	44	
51	52	53	54	

As before, in this basic format, it is assumed that the data is sample-oriented, (*i.e.*, one sample's information is tabulated completely before the tabulation of the next sample begins).

To include variable names when sample names are unavailable, the following format may be best:

Table 14.4
ASCII file with
grouped column
labels

#d	4x5			
#c	V1	V2	V3	V4
11	12	13	14	
21	22	23	24	
31	32	33	34	
41	42	43	44	
51	52	53	54	

If you want to include both sample names and variable names, put the variable (column) names first, and Pirouette will expect the data to be organized by samples. This format is demonstrated next.

Table 14.5
ASCII file with
grouped column and
row labels

#d	4x5				
#c	V1	V2	V3	V4	
#r	S1	S2	S3	S4	S5
11	12	13	14		
21	22	23	24		
31	32	33	34		
41	42	43	44		
51	52	53	54		

Data can be also organized such that no variable names are supplied and the sample name is included with the sample's data, as in the following table.

Table 14.6
ASCII file with
sample names in
each row

#d	4x5				
#s	S1	11	12	13	14
#s	S2	21	22	23	24
#s	S3	31	32	33	34
#s	S4	41	42	43	44
#s	S5	51	52	53	54

Adding variable names to the above format requires that they be grouped after the #c specifier:

Table 14.7
ASCII file adding
variable names

#d	4x5				
#c	V1	V2	V3	V4	
#s	S1	11	12	13	14
#s	S2	21	22	23	24
#s	S3	31	32	33	34
#s	S4	41	42	43	44
#s	S5	51	52	53	54

If your data is organized by variable rather than by sample, Pirouette will read the transposed file whenever the #r specifier precedes the #c specifier, as shown in the next table. This means that values in rows in the ASCII file become values in columns in Pirouette.

14 Data Input: Common File Formats

Note that the files in [Table 14.8](#) and [Table 14.6](#) will produce the same Pirouette spreadsheet.

Table 14.8
Transposed ASCII
with row labels

#d	4x5				
#r	S1	S2	S3	S4	S5
11	21	31	41	51	
12	22	32	42	52	
13	23	33	43	53	
14	24	34	44	54	

Similarly, if you list the row (sample) names first then the variable names, the ASCII file read assumes that the data is organized by variables:

Table 14.9
Transposed ASCII
with column and row
labels

#d	4x5				
#r	S1	S2	S3	S4	S5
#c	V1	V2	V3	V4	
11	21	31	41	51	
12	22	32	42	52	
13	23	33	43	53	
14	24	34	44	54	

Data can also be organized such that the variable name is included with the variable's data:

Table 14.10
Transposed ASCII
file organized by
variables

#d	4x5					
#v	V1	11	21	31	41	51
#v	V2	12	22	32	42	52
#v	V3	13	23	33	43	53
#v	V4	14	24	34	44	54

To add the sample names to the above format requires that they be grouped after the #r specifier:

Table 14.11
ASCII file adding
sample names

#d	4x5					
#r	S1	S2	S3	S4	S5	
#v	V1	11	21	31	41	51
#v	V2	12	22	32	42	52
#v	V3	13	23	33	43	53
#v	V4	14	24	34	44	54

Up to this point all data values have been assumed to be independent variables. Markers can be added to the ASCII file to indicate the type of the variables or to define a missing value:

- * preceding a variable name indicates a class variable
- \$ preceding a variable name indicates a dependent variable
- M indicates a missing value for that data value

The next table shows a file with 3 samples, each with 4 variables. All sample and variable names are included. In addition, variable AA has been assigned as a class variable, and Sr as a dependent variable. There is one missing value.

Table 14.12
ASCII file with
dependent and class
variables

#D	4x3				
#C	*AA	Ca	Ba	\$Sr	
#S	SA1	12	16	15	18
#S	SA2	14	19	11	M
#S	SA3	14	11	17	19

Note: When using #v or #s specifiers to indicate names and variable values or names and sample values, it is not necessary that every variable contain the #v plus name combination. The #v and name can be left off of the beginning of that variable's data; it will still be read properly. Similarly, the #s and name can be left off of the beginning of the list of the data values for samples. When reading in ASCII data, be sure that the dimensionality specifier accurately reflects the number of samples and variables.

Note: You cannot have spaces in variable or sample names because the space is interpreted as a field delimiter. Instead, use a character, such as the underscore, to imply a space in a name.

EXCEL FILES

Importing Excel files is a simple procedure because, with few exceptions, Pirouette imports them directly. Both formats follow a pattern similar to ASCII files, giving you the ability to import data with or without labels. However, it is possible to read in Excel files with labels for samples but not for variables and vice versa. In every instance, the first cell, *i.e.*, row 1 and column 1, should be left blank.

Pirouette assumes that data is in a rectangular format and that variables are stored column-wise and samples are stored row-wise. Also, variable names can only occupy the first row of the spreadsheet beginning with cell B1—the variable labels are at the top of the columns containing the values for the variable they represent. Sample names must be located in the first column beginning in cell A2, and correspond to sample data across the row. Labels will be treated as text when read into Pirouette.

If you wish to read in an Excel data file without sample or variable names, the corresponding column (A) or row (1) must be left blank. Additional Excel file specifications follow:

- Use an M (or m) in all cells with missing values. Missing values will have cells filled with asterisks (*) in the Pirouette spreadsheet.
- A variable will be read and assigned as a class variable if the variable label in row 1 is preceded by an asterisk. For example, “*Process” would be read in as a class variable.
- A variable will be read and assigned as a dependent variable if the variable label in row 1 is preceded by a dollar sign. For example, “\$Octane” would be read as a dependent variable.

Other File Formats

All remaining file formats currently supported by Pirouette are described in this section. The origin and purpose of each format are described, and, where necessary, tips on the use of the originating software are given to make the conversion as smooth as possible. Be aware that as new file read modules are developed, we make them available in future releases and/or place them on the [Infometrix web site](#).

Note that the chromatography peak data formats are mostly present in two forms: one will read peak areas; the other will read peak heights. You can choose to save your data including one or the other or both; be sure that the data type is present when you then use Pirouette to read the data.

Agilent ChemStation

*.ch

The Agilent (formerly Hewlett Packard) ChemStation line of chromatographs use a common file format for GCs, LCs and other instruments. The actual data file has .CH as the extension, but is embedded within a directory which carries the name of the sample. And, the directory name itself has an extension .D. Other data, including peak reports and other sample parameters, are stored as individual files in the same sample directory.

Note: *Although employing the same name, this format is not to be confused with the ChemStation format used with the mass spectrometer systems. These use similar .D sample directories but the GC-MS data are stored in a .MS file.*

ASD Indico Pro

*.asd

The Indico Pro format is used by ASD for their LabSpec and QualitySpec analyzers, as well as their FieldSpec 3 and Pro version instruments. These are binary files.

AIA or ANDI

*.cdf

Variations include:

- All Peak Areas or Heights
- AIA Named Peak Areas or Heights
- AIA Raw Data
- AIA Data with RT Markers

In its simplest form, the AIA Standard is a “generic system for analytical data interchange and storage.” Adopted by members of the Analytical Instruments Association (AIA), it is in the process of being implemented in all major chromatography software systems.

In Pirouette, the AIA format comprises two categories of information common to chromatographic data: Raw Data and Final Results. The Raw Data format will extract the actual chromatographic profile from a .CDF file. The Data with RT Markers format will

also load the whole chromatographic profile but, in addition, will search the AIA file for named peaks and enter the corresponding scan times of these peaks into Y variable columns. This is a convenience for those needing to pretreat chromatographic data by alignment (see “Align” on page 4-22).

The latter category includes the “amounts and identities (if determinable) of each component in a sample.” The Pirouette converter allows the Final Results to be extracted as either the peak areas or the peak heights; you also can choose whether to extract all the peak values or only those for which a “name” has been applied by the chromatographic software.

The user of an AIA-supporting chromatography data system must save the analysis results in an AIA-type file. The extension of this file should be .CDF, which is used by Pirouette as the file filter. The means to save such a file will vary with the software; it may be an integral part of the software or it may be a part of a separate file converter.

Analect FT-IR

***.asf**

Analect makes a high-resolution FTIR spectrophotometer whose files can be read by Pirouette. Spectra are stored as individual files with a .ASF extension and are written into binary format files.

Brimrose AOTF

***.dat**

Brimrose makes an AOTF spectrophotometer known as the Luminar 2000. Brimrose data files are written in binary format and may contain single or multiple spectra. The spectra may be the raw spectra or they may be a transformation of the raw data (e.g., derivative spectra).

Thermo Scientific GRAMS

***.spc**

GRAMS (and its sister programs Spectra Calc and Lab Calc) are software packages mainly intended for the spectroscopist. The GRAMS file structure is a binary format which can contain either single or multiple spectra (or chromatograms). These files can be read directly by Pirouette. In addition, through its setup procedure, GRAMS can convert many other formats to its own.

Guided Wave Models 300, 310 and 412

***.asc, *.*ed**

The Guided Wave Model 412 is a process near infrared spectrophotometer. Spectra collected by the M412 can be stored as single spectra files or as spectral files that include lab values. The latter files can have extension .updated or .completed. The *.*ED filter will allow both of these file types to be displayed simultaneously. Data collected from the older Models 300 and 310 can be displayed with the *.ASC filter.

Agilent (Hewlett Packard) Model 8453

***.wav**

The Model 8453 is a UV/VIS diode array spectrophotometer, collecting data from 190 to 1100 nm. Model 8453 data spectra are written into ASCII format as single spectrum files.

JCAMP-DX Infrared Spectroscopy

***.dx, *.jdx**

JCAMP is an attempt at a standardized spectroscopy data interchange format. The generic nature of the format allows it to be used for any data type, but its primary use has been in IR spectroscopy. Most major IR vendors support JCAMP by allowing the user to export to a .DX file. However, variations in the implementation of the JCAMP standard do exist, and it is possible that Pirouette may not recognize a particular variant. At present, Pirouette is designed to read valid formats up to version 4.24.

LT Industries Model 1200

***.dat**

The Model 1200 is an infrared spectrophotometer sold by LT Industries. Model 1200 files are stored in binary format and can contain multiple spectra in a single file. If the .ING file, containing dependent variable information, is also present, these data will be appended to the Pirouette spreadsheet.

NWA Quality Analyst

***.nqa**

Northwest Analytical makes software for MSPC. Data files can be exported from their package as a custom text format file and imported into Pirouette for analysis.

FOSS NIRSystems NSAS

***.da**

NSAS data files are written in binary format and may contain as many spectra as there is room for on the disk. The spectra may be the raw spectra or they may be a transformation of the raw data (*e.g.*, derivative spectra). If the .CN file, containing dependent variable information, is also present, these data will be appended to the Pirouette spreadsheet.

Perkin-Elmer Spectrum for Windows

***.sp**

Perkin Elmer makes several spectrophotometers which can save either single IR spectra and/or single interferograms in an ASCII file. The Pirouette file server reads older and newer versions of spectral files saved in this format.

Bruker OPUS

***.0**

The OPUS format developed by Bruker will save various types of spectra. The default extension is “.0”, which is what is shown in the Pirouette file filter. Bruker will save subsequent file copies with increments in the extension (.1, .2, etc.). To load these into Pirouette, select the OPUS file type, change the File Name filter from *.0 to *.* , and hit enter to list the additional spectra.

AIT PioNIR

***.pdf**

PIONIR is a near-infrared spectrometer that provides on-line monitoring of liquid streams in refining and petrochemical processes.

Output of Results

Contents

Printing	15-1
Capturing Chart Windows	15-2
Saving Files	15-3
Saving Models	15-6

Pirouette provides several options for handling results generated within the program, including printing, capturing of graphic images, and saving data files and/or computed results and saving multivariate models. The various functions are all accessed from the File menu and are described in this chapter.

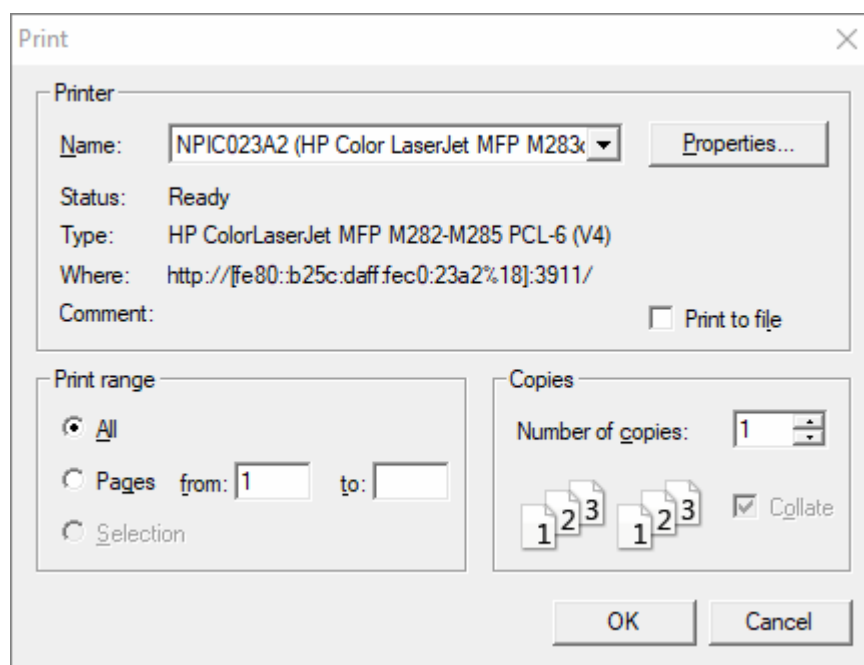
Printing

Pirouette output can be printed in two different ways. It can either be sent directly to your printer or it can be saved in a file for printing later. Choose the Print Setup item on the File menu to select and configure a printer. The printers available from Print Setup are those that have been previously installed on your system.

Be sure that your destination printer is powered on and on line before you begin to print. To print the contents of the current chart window,

- Select Print from the File menu (or use the Ctrl-P shortcut)

Figure 15.1
The print dialog box



The example print dialog box shown above allows you to specify the number of copies desired and whether you want to print to a file.

Capturing Chart Windows

To capture graphics in a file for incorporation into a report or other document, you can copy the image into the Windows clipboard as a bitmap or as a metafile. To use this feature,

- Click on the window you wish to capture to make it current
- Select Copy from the Edit menu, to capture a bitmap (TIFF format), or
- Select Copy Special/To Clipboard, from the Edit menu, to capture a metafile

Any Pirouette chart can be captured in this manner if its window is first made current. This is true, however, only for chart windows containing a single plot; chart windows containing an array of subplots can only be captured as bitmaps.

With this copy command, the window contents will be copied. If you also want to retain the window's title bar, hold the shift key down and select the copy menu item.

A window with a data table can be copied as a bitmap as well. First, however, make sure that no cells are highlighted because the copy command puts the contents of highlighted cells on the clipboard; see [“Cut, Copy, Paste, and Clear” on page 13-9](#). You can insure that no cells are highlighted by clicking in an empty cell, outside of the data range or just by insuring that the active cell is not in the viewing window when you do the copy. Note that a bitmap copy of a data table will include only that portion of the table which is displayed in the window.

When choosing to copy a metafile, you will copy the graphic as an Enhanced metafile (EMF which has a richer set of drawing entities compared to the older WMF format).

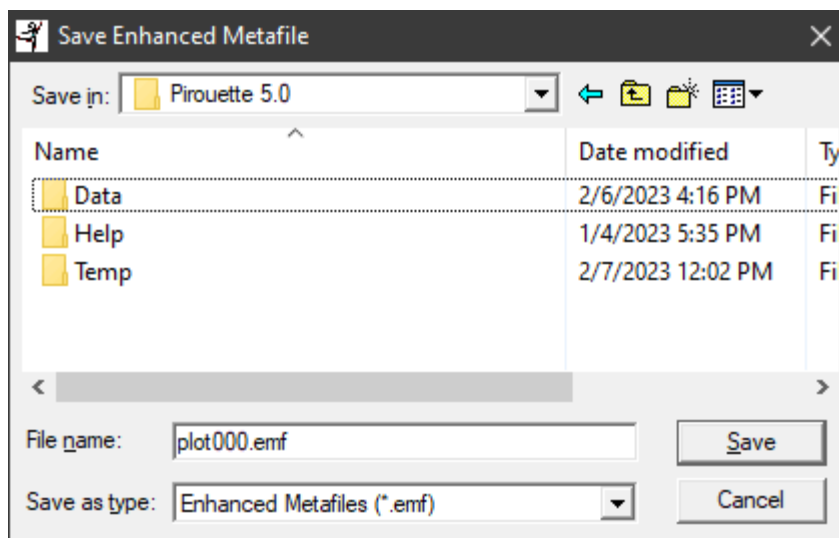
After the copy, you can paste the image into a document open in another application. You can also save your image to a file with the Clipboard Viewer accessory or a third party utility such as SnagIt, which saves images in a number of common graphic formats. The clipboard retains the copied image until the next copy operation.

Alternatively, you can save the EMF to file.

- Choose Edit > Copy Special > To File

and a dialog will be present for you to name the file and choose its destination.

Figure 15.2
Saving an EMF
format graphic



A default name will be provided (such as plot000.emf), otherwise you should enter a meaningful name to help keep track of results.

Saving Files

Pirouette supports several output formats.

- Pirouette
- ASCII
- Excel
- Flat ASCII
- Galactic SPC
- ChemStation CH
- AIA/ANDI CDF

Pirouette files are saved in either PIR2 format or as legacy PIR files. Unless you are supporting applications using older versions of Pirouette, we suggest staying with the more capable PIR2 format. Saving data in the Pirouette format stores not only the raw data but also subsets, algorithm results and models in a single file accessible only by Pirouette or another Infometrix package or client.

The Excel and ASCII output formats store just the raw data, preserving the variable type (*i.e.*, independent, dependent, and class) and names of samples and variables. Data saved

15 Output of Results: Saving Files

in the Flat ASCII format contain only the rectangular matrix of data; no sample or variable names are retained. This format facilitates loading of data into math packages such as Matlab.

Note: *Scripts for Matlab are available for loading from and saving to the Pirouette ASCII (*.DAT) format.*

Finally, Pirouette data can be stored in either of the two most common data interchange formats, .SPC and .CDF. The SPC files are stored in the multifile format, that is, all spectra are stored in a single file. On the other hand, the CDF files are stored as single sample files. To avoid confusion, the saved files have a serialized suffix added to each name before the extension.

Computed results are not stored together with the raw data in any of these formats except the Pirouette binary format. However, most computed objects can be saved independently into its own separate file (see below).

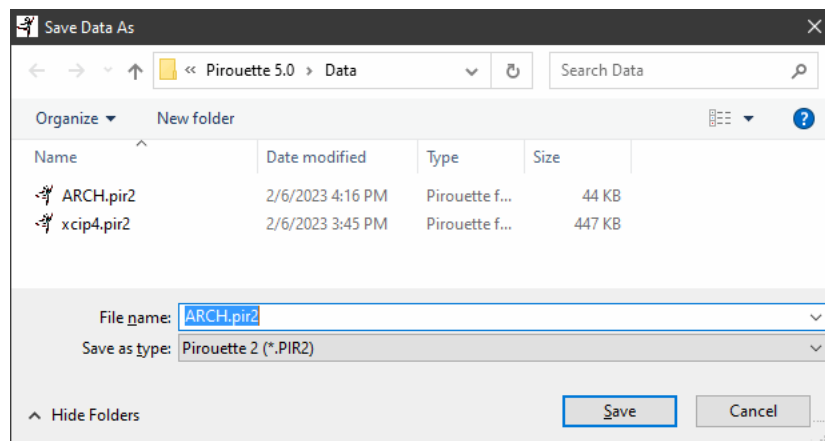
Note: *Older versions of Excel allow loading only 256 columns into a spreadsheet. Thus, any file saved in this format and containing more than 256 variables will be truncated when read into the program.*

SAVING DATA

To save the raw data (and all existing subsets and results) in the case of a Pirouette binary format),

- Choose Save Data As from the File menu

Figure 15.3
Save Data As dialog
box



The dialog box which will open lets you specify the name of the file to be saved, the file type, and its location:

- Navigate to the file's destination directory and drive
- Set the output format via Save as type
- Type a file name or click on an already existing name
- Click on OK

A suggested file name will be supplied, based on the name of the file currently in use.

Note: *Because the output format is determined by the Type of File setting and not the extension assigned to the file, you can override the default extensions supplied in Pirouette. However, it may later be difficult to determine the format of the file.*

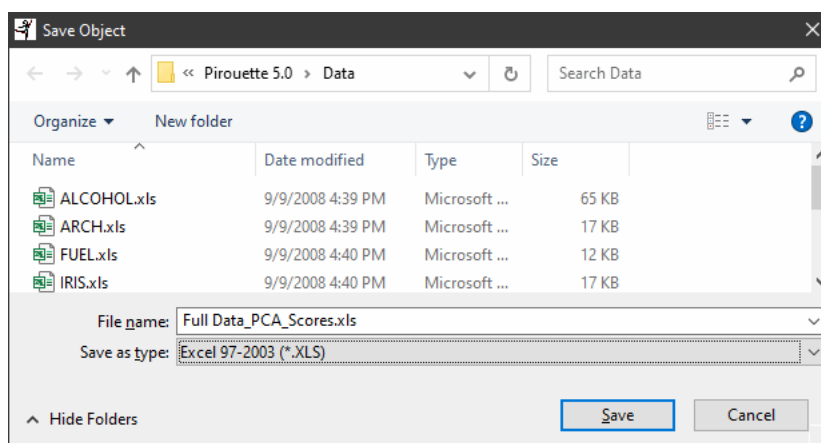
SAVING RESULTS

To save just the included data in an exclusion set or an algorithm result,

- Choose Save Object from the File menu

The dialog box shown below is similar to the Save Data As dialog box in [Figure 15.3](#), except that the suggested name will identify the specified object.

Figure 15.4
Save Objects dialog
box



- Navigate to the desired directory and drive
- Set the output format via the File Type
- Accept the suggested file name, type a new file name or click on an already existing name
- Click on OK

Objects saved in an ASCII format can be read into any text editor. Objects saved in Excel format can be imported into spreadsheets that support the XLS format, where you may want to perform some special processing or plotting. Because Pirouette can read these formats, you can load or merge the object file back into Pirouette itself.

Note: *Objects reloaded into Pirouette are treated as raw data. The object's original identity is lost; it is merely a table of numbers. The stored row and column names may help you deduce the source of the data but a descriptive file name is advised.*

It is also possible to save data into an AIA format. Because this format is found in many chromatographic data systems, Pirouette results could be loaded into a chromatographic system for further processing. This is useful for data which has been aligned by the Pirouette transform of the same name (see [“Align” on page 4-22](#)).

The AIA format only permits a single sample per file, thus when Pirouette saves a multi-sample data object to this format, it will serialize the resulting file names by adding a suf-

fix to the name you supply, where the suffix is a serial number starting at X2 (each new file save restarts the serial number at X2), where X represents the number of padding zeros. Zeros are added only to make each name unique and alphabetizable. Thus, if there are 123 samples, then there would be padded leading zeros as needed to make the suffix contain 3 digits.

Saving Models

All algorithms in Pirouette except HCA produce models. PCA, an exploratory algorithm, produces models which are then used to compare future data by projection into the principal component space of a training set of data. A classification model (resulting from either KNN, SIMCA, or PLS-DA) is used to predict a future sample's category. This may be as simple as classifying samples as good or bad or as complex as a model created for the Centers for Disease Control which categorizes over thirty types of mycobacteria, including *M. tuberculosis*. Regression models (produced by PCR or PLS) typically predict a property or concentration, such as the octane number of gasoline from an analysis of NIR spectra. PLS-DA is a hybrid: it uses the core PLS algorithm to perform classification.

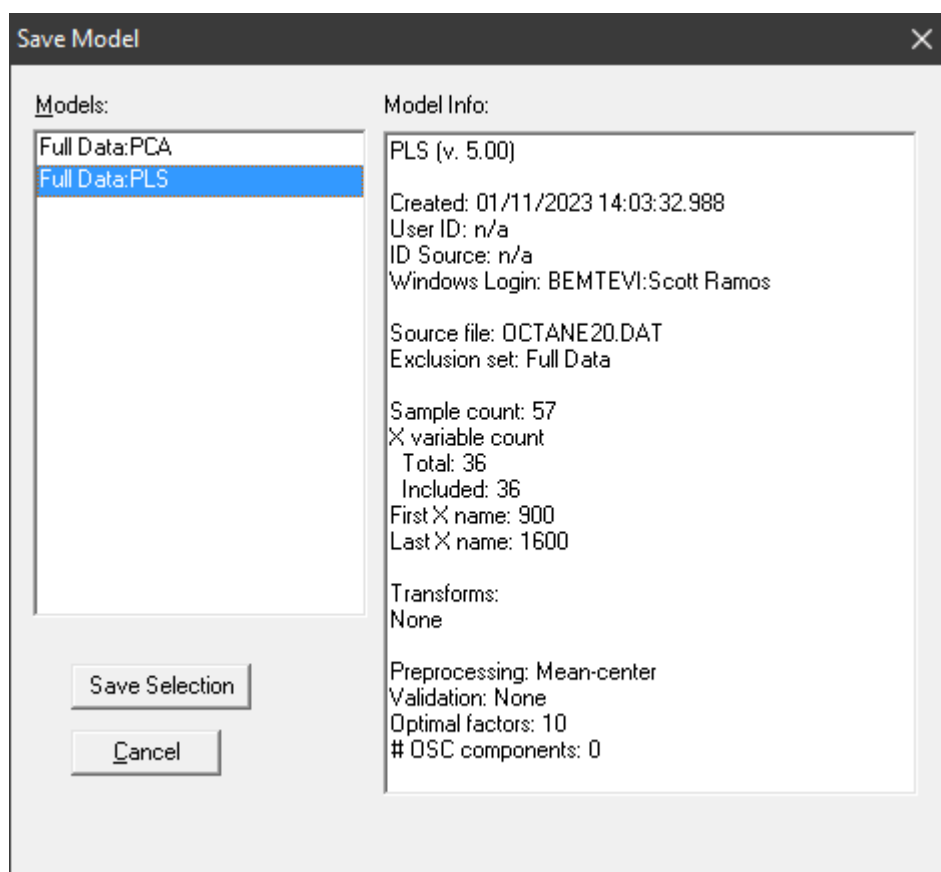
PIROUETTE MODELS

Whenever a modeling algorithm runs, an algorithm-linked model is produced. The model persists until the algorithm run that generated it is deleted. The model can be unlinked from the algorithm run and stored into a file. Saving a model in Pirouette format allows you to open it later and make the prediction in either Pirouette or another Infometrix product such as InStep. To save a model,

- Choose the Save Model menu item in the File menu

which opens a dialog box like that shown next.

Figure 15.5
Selecting a Model to
save



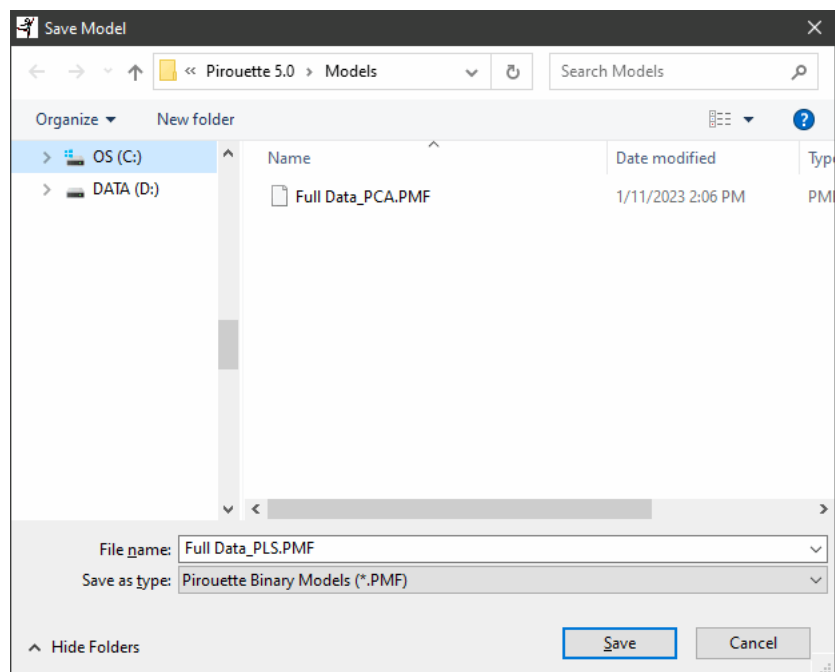
As you highlight one from a list of available models, information about it appears in the Model Info box (on the right side of the dialog box).

- Click on the Save Selection button
- Navigate to the desired directory and drive
- Set the Model Type (if a choice is available)
- Type a file name or click on an already existing name
- Click on Save

The dialog box will appear as shown in Figure 15.6 below.

15 Output of Results: Saving Models

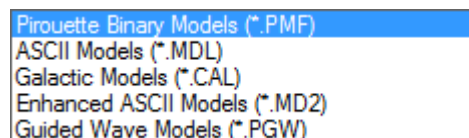
Figure 15.6
Save Model dialog
box



Note: *InStep and Pirouette cannot open .MDL or .MD2 type models so save in these ASCII formats only if you plan to later access the model with custom software or a macro. See the next section for information on ASCII Models.*

Note that there are several formats available.

Figure 15.7
List of model types



Each of these model types serves a different purpose, as noted below.

Table 15.1
Pirouette model
formats

Name	Extension	Use
Pirouette	*.PMF	Predictions within Pirouette version 2.0 or later; and with InStep versions 2.0 and newer. All model types.
ASCII	*.MDL	(see below). PLS and PCR models only.
ASCII enhanced	*.MD2	Similar to ASCII, with extra information on new transforms. PLS, PCR, KNN and SIMCA models contain additional component pieces (see below)
Galactic	*.CAL	Emulates SpectraCalc/LabCalc calibration files. PLS and PCR models.
Guided Wave	*.PGW	Emulates Guide Wave calibration files. PLS models only.

ASCII MODELS

Relying on Pirouette or InStep to make predictions keeps life simple. However, sometimes model creation occurs in Pirouette but predictions must occur outside the Pirouette or InStep environment. PCR and PLS models consist mostly of a regression vector and data processing parameters. Choose the .MDL file type in the Save Models dialog box to save a regression model in an ASCII format. This rudimentary format does not support all of Pirouette's transforms. Thus, its utility is limited to predicting a y value for models that use the so-called legacy transforms, those available in DOS Pirouette.

The ASCII model includes the name of the file from which the model was derived, transforms, preprocessing options, offset and factor vectors (for preprocessing) and, finally, the regression vector. If more than one dependent variable was included, each dependent variable model is listed, in sequence. Table 15.2 shows a PLS ASCII model file created from DAIRY.DAT, which is furnished with Pirouette.

Table 15.2
An ASCII model file,
.MDL format

```

Calibration Method: PLS
Original data file:  DAIRY
Number of dependent variables: 2
Number of variables: 6
Transforms: Smoothing(5), Normalize (100.00)
Scaling: Autoscale
Validation: None
Number of Leave Out: 1

X-Offset  ··  2.32764e+00  ··  3.37298e+00  ··  8.97816e+00  ··  1.91064e+00  ··  3.18654e+01
           ··  4.13826e+01
X-Factor  ··  6.06414e+00  ··  6.03093e+00  ··  5.04562e+00  ··  3.10448e+00  ··  7.88556e-01
           ··  1.33903e+00
Y-Offset  ··  4.17006e+00  ··  2.74836e+01
Y-Factor  ··  1.07526e+00  ··  7.70650e-01

Dependent variable: Fat
Range: 3.87000e+01 to 4.42000e+01
Optimal number of components: 5
Standard error: 5.54626e-01
Regression vector  ··  -3.29700e+00  ··  -2.05223e+00  ··  -1.16270e+00  ··  2.57833e+00  ··
                   2.63031e+00  ··  -2.43671e+00

Dependent variable: Moisture
Range: 2.60000e+01 to 3.20000e+01
Optimal number of components: 5
Standard error: 4.56424e-01
Regression vector  ··  3.313987e+00  ··  2.08165e+00  ··  1.29630e+00  ··  -1.10915e+00  ··
                   2.74743e+00  ··  2.77295e+00

```

Beginning with Pirouette 3.0, enhanced ASCII formats (.MD2) for PLS and PCR became available. They contain information necessary to implement non-legacy transform methods like MSC, DivideBy, etc.

For the current version (and dating to Pirouette version 4.0), the enhanced ASCII versions for PLS and PCR were significantly modified to permit the computation of several outlier diagnostics and the application of OSC. The following section provides guidance to 3rd party developers attempting to implement prediction from these enhanced ASCII models. The following table uses the same data as Table 15.2 but shows additional items.

Table 15.3
An enhanced ASCII
PLS model file, .MD2
format

```

Calibration Method: PLS
Original data file: DAIRY.DAT
Number of dependent variables: 2
Number of variables: 14
Number of model samples: 140
Transforms: MSC
Preprocessing: Mean-center
Validation: Cross(1)
Number of OSC components: 1

Inclusion bitmap: 1 .. 1 .. 1 .. 1 .. 1 .. 1 .. 1 .. 1 .. 0 .. 0 .. 0 .. 0 .. 0 .. 0 .. 0 .. 0
X-Offset: 5.35939e-02 .. 1.74324e-01 .. 2.72141e-01 .. 3.96485e-01 .. 8.83203e-01 .. 9.89484e-01 ..
1.02481e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00
X-Factor: 1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 ..
1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 .. 1.00000e+00 ..
1.00000e+00 .. 1.00000e+00
Y-Offset: 4.17006e+01 .. 2.74836e+01
Y-Factor: 1.00000e+00 .. 1.00000e+00

Dependent variable: Fat
Range: 3.87000e+01 to 4.42000e+01
Optimal number of components: 1
Standard error: 4.25595e-01
Model residual: 3.02518e-03
Regression vector: -3.33699e+01 .. -8.50063e+01 .. 5.36245e+01 .. 1.09024e+02 .. -2.40354e+01 .. -
2.00827e+01 .. -1.53298e-01 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00 .. 0.00000e+00
Loading1: -2.15033e-01 .. -5.47774e-01 .. 3.45552e-01 .. 7.02539e-01 .. -1.54882e-01 .. -1.29411e-01 .. -
9.87842e-04 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00
Model scores mean: -3.36357e-09
Scores covariance inverse: 2.46277e+04
OSC Weight1: -1.25501e-01 .. 1.29257e-01 .. -4.78947e-01 .. 4.64790e-01 .. 4.85006e-01 .. 3.31443e-01 ..
-8.06044e-01 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00
OSC Loading1: -1.54591e-01 .. -2.43830e-01 .. -7.37004e-02 .. 5.39107e-01 .. 5.26820e-01 .. -6.94291e-03 ..
-5.86861e-01 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00

Dependent variable: Moisture
Range: 2.60000e+01 to 3.20000e+01
Optimal number of components: 2
Standard error: 3.21741e-01
Model residual: 2.79886e-04
Regression vector: 2.05246e+00 .. 7.37562e+01 .. -2.25461e+01 .. -7.86170e+01 .. 8.77612e+00 ..
1.06260e+01 .. 5.95180e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00
Loading1: 7.89303e-02 .. 6.35015e-01 .. -2.51649e-01 .. -7.09890e-01 .. 1.04451e-01 .. 1.03979e-01 ..
3.91596e-02 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00
Loading2: -6.82458e-01 .. 3.50742e-01 .. 5.42922e-01 .. -6.72590e-04 .. -2.84742e-01 .. -9.06361e-02 ..
1.64841e-01 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00
Model scores mean: 9.18631e-09 .. -2.52530e-09
Scores covariance inverse: 2.44658e+04 .. 2.16515e+03 .. 2.16515e+03 .. 5.01398e+04
OSC Weight1: -3.86225e-01 .. 1.71878e-01 .. 1.66720e-02 .. 1.74333e-01 .. 7.59311e-01 .. -1.84931e-02 ..
-7.17478e-01 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00
OSC Loading1: -3.21133e-01 .. -1.20419e-01 .. 1.03344e-02 .. 5.34355e-01 .. 5.08441e-01 .. -3.07851e-02 ..
-5.80792e-01 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 .. 0.00000e+00 ..
0.00000e+00 .. 0.00000e+00

MSC Ideal: 5.34334e-02 .. 1.74245e-01 .. 2.72282e-01 .. 3.96690e-01 .. 8.83137e-01 .. 9.89438e-01 ..
1.02482e+00 .. 9.43797e-01 .. 8.83227e-01 .. 8.02673e-01 .. 6.16289e-01 .. 4.67190e-01 .. 2.29000e-
01 .. 3.04064e-01
    
```

In [Table 15.4](#) the PLS model file contents are related to quantities previously described in this user guide.

Table 15.4
Decoding an
enhanced ASCII PLS
model

Model Field	Comments	Symbol
Number of dependent variables	# of Ys in the model	n_Y
Number Of Variables	# of independent (that is, X) variables	m
Number Of Model Samples	# of samples in model's training set	n_{model}
Number Of OSC Components	# of OSC components retained in the model	k_{osc}
Inclusion bitmap	A m element vector of 1s and 0s; 1s indicate included X variables	
X-Offset, X-Factor	Two m element scaling vectors for the x block	
Y-Offset, Y-Factor	Two n_Y element scaling vectors for the y block	
Optimal Number Of Components	# of PLS components retained in the model	k
Regression Vector	An m element vector	β
Loading1, ...Loadingk	The k vectors comprise \mathbf{L} , the loadings matrix	\mathbf{L}
OSC Weight1, ...OSC Weight k_{osc}	The k_{osc} vectors comprise \mathbf{W}_{osc} , the matrix of OSC weights	\mathbf{W}_{osc}
OSC Loading1, ...OSC Loading k_{osc}	The k_{osc} vectors comprise \mathbf{L}_{osc} , the matrix of OSC loadings	\mathbf{L}_{osc}
Model scores mean	A k element vector containing the means of the training scores	$\bar{\mathbf{t}}$
Model residual	The sum of sum of squares of the training set's unscaled X residuals	$\text{ESS}_{\text{model}}$
Scores covariance inverse	A square matrix with k rows and columns	\mathbf{s}^{-1}

Calculations with an ASCII model

The following discussion assumes a single sample vector with m elements, \mathbf{x}_{unk} , and model with a single Y .

1. Load the file and compose matrices from the vectors stored in it.

At minimum, assemble the k loading vectors, which contain m elements, into a matrix \mathbf{L} with m rows and k columns. If the Number of OSC Components is greater than 0, two other matrices, \mathbf{W}_{osc} and \mathbf{L}_{osc} , must also be assembled in the same manner as \mathbf{L} . Both OSC matrices contain m rows and k_{osc} columns.

The Scores covariance inverse field is followed by $k*k$ values. These must be shaped into a k by k matrix in order to compute the Mahalanobis distance of the unknown. The first k values comprise the first row of the matrix, the next k values comprise the second row, etc.

2. Apply transform(s)

This step is necessary only if the Transforms field value contains something besides None; in which case the methods and parameters are listed. Note that some transforms require the entire x vector (*e.g.*, derivatives) while others operate from only included

15 Output of Results: Saving Models

variables. Code is not given here for each transform; see “Transforms” on page 4-10 for details. If more than one method is listed, they must be applied in the order listed.

Note: *Some transform methods are simple to implement after a careful reading of this manual. Others are more challenging. Forewarned is forearmed!*

3. Apply bitmap

Multiply element-by-element the values in \mathbf{x}_{unk} and the Inclusion bitmap. This essentially zeroes out excluded X variables in subsequent computations using full-width vectors and matrix.

4. Preprocess

Subtract X_Offset element-by-element from \mathbf{x}_{unk} , then do an element-by-element division of the difference by X_Factor . This produces the preprocessed (or scaled) \mathbf{x}_{unk} .

5. Orthogonalize

This step is necessary only if OSC was included as part of the PLS calibration, that is, if $k_{\text{osc}} > 0$. If so, the preprocessed \mathbf{x}_{unk} must be orthogonalized before prediction:

$$\mathbf{x}_{\text{unk}} = \mathbf{x}_{\text{unk}} - (\mathbf{x}_{\text{unk}} * \mathbf{W}_{\text{osc}}) * \mathbf{L}_{\text{osc}}^T$$

6. Compute Y

The scaled predicted Y value is computed from \mathbf{x}_{unk} and the regression vector β :

$$\mathbf{y}_{\text{scaled}} = \mathbf{x}_{\text{unk}} * \beta^T$$

To get an unscaled value, multiply $\mathbf{y}_{\text{scaled}}$ by Y_Factor and add Y_Offset . This essentially undoes the preprocessing.

7. Compute Mahalanobis Distance

The Mahalanobis distance (MD) depends on the mean -centered prediction scores which can be computed from \mathbf{L} , the \mathbf{x}_{unk} vector from step 6, and from $\bar{\mathbf{t}}$.

$$\mathbf{t}_{\text{unk}} = (\mathbf{x}_{\text{unk}} * \mathbf{L}) - \bar{\mathbf{t}}$$

The Scores covariance inverse assembled in step 1 is now necessary:

$$MD = \mathbf{t}_{\text{unk}} * \mathbf{S}^{-1} * \mathbf{t}_{\text{unk}}^T$$

To compute the MD threshold, consult a table of Chi squared values, specifying the number of model factors k and a prediction probability.

8. F Ratio

To compute the F Ratio, first compute \mathbf{e} , the X residuals vector.

$$\mathbf{e}_{\text{unk}} = \mathbf{x}_{\text{unk}} - \mathbf{x}_{\text{unk}} * \mathbf{L} * \mathbf{L}^T$$

Because the PLS model contains the **unscaled** model residual, the unknown's X residuals must also be unscaled by multiplying the vector element-by element by X_Factor . Then each element of the unscaled X residuals is squared and summed to get a scalar, the unknown's error sum of squares, ESS_{unk} . The desired quantity, the F ratio, is the ratio of this value to the **average** model residual:

$$F = ESS_{\text{unk}} / (ESS_{\text{model}} / n_{\text{model}})$$

The F ratio probability, P , can then derived from F . F_{prob} is a function which calculates the probability value from an F value and two degrees of freedom parameters:

$$P = 1 - \text{Fprob}(F, df1, df2)$$

where $df1 = 1$ and $df2 = n_{\text{model}} - k$. Information on computing an F-distribution probability function¹ can be found in many standard statistical and numerical analysis texts.

The F Ratio threshold may be found by consulting a table of F values for the specified prediction probability. The lookup depends on two degrees of freedom: $df1 = 1$ and $df2 = n_{\text{model}} - k$. If the model contains a preprocessing option that involves subtracting a mean and n_{model} is less than or equal to m , $df2 = n_{\text{model}} - k - 1$.

Note: Coding strategies for KNN and SIMCA enhanced ASCII models are not detailed here. For information about using these models to make predictions, [contact Infometrix](#).

GALACTIC MODELS

For PLS and PCR, models can be saved in a format compatible with SpectraCalc/Lab-Calc. Not all Pirouette generated PLS and PCR models can be saved in the .CAL format. For example, a Galactic model must specify a preprocessing setting of either mean-center, autoscale, or variance scale. Moreover, it cannot have any transforms, all x variable names must be numeric, and the number of OSC components must be 0. Pirouette models with multiple Ys produce a series of single-Y Galactic model files.

Note: As of Pirouette v. 4 CAL models saved from validated PLS and PCR runs contain the validated ESS in the model residual field. Previously the unvalidated ESS was written out.

References

1. Press, W.H.; Flannery, B.P.; Teukolsky, S.A. and Vetterling, W.T.; *Numerical Recipes* (Cambridge University Press, Cambridge, 1986), p. 169.
Abramowitz, M. and Stegun, I.A.; *Handbook of Mathematical Functions*, NBS Applied mathematics Series 55 (National Bureau of Standards, Washington DC, 1967), p. 946.


Pirouette Reference

Contents

Menu Features and Shortcuts	16-1
File Menu	16-3
Edit Menu	16-11
Process Menu	16-19
Display Menu	16-34
Objects Menu	16-38
Windows Menu	16-39
Help Menu	16-45

This chapter lists every menu item in Pirouette. A brief description and a reference to a more focused discussion is supplied. About half of Pirouette's menu items are found in virtually all Windows programs. Whenever possible, we adhere to the standards established by Microsoft and Apple in their Windows and Macintosh product lines. Therefore, common functions are discussed briefly, elaborating only where a special behavior exists within Pirouette.

Menu Features and Shortcuts

Menus can be accessed either with the mouse or from the keyboard. Keyboard access is discussed below. An ellipsis following a menu item means that a dialog box will be displayed when the option is selected. Any menu item followed by the triangle symbol () has an associated pull-down submenu.

Using the keyboard, you can open a menu or select an item from an open menu by pressing the underlined letter associated with the item while holding down the Alt key. The Alt key is optional for selecting an item from an open menu.

In addition, single stroke keyboard equivalents allow you to access a menu item without first opening its menu. To execute the keyboard shortcut, press the key combination displayed to the right of the item. Shortcuts are listed in the following table. Shortcut availability in the menus are context-sensitive and show up when an object can benefit from their use.

Table 16.1
Keyboard Shortcuts

Menu	Menu item	Equivalent
File	Open Data	Ctrl+D
	Save Data	Ctrl+S
	Open Model	Ctrl+M
	Print	Ctrl+P
	Quit	Alt+F4
Edit	Undo	Ctrl+Z
	Cut	Ctrl+X
	Copy	Ctrl+C
	Paste	Ctrl+V
	Delete	Del
	Activate Class	Ctrl+K
	Exclude	Ctrl+E
	Include	Ctrl+I
Process	Run	Ctrl+R
	Predict	Ctrl+Y
Display	Selector	F12
	Zoom Current Plot	Enter
	Unzoom Current Plot	Ctrl+Enter
	Limits	Ctrl+B
	Labels	Ctrl+L
Objects	Find	Ctrl+F
	Rename	Ctrl+N
Windows	Cascade	Shift+F5
	Tile	Shift+F4
	Close Window	Ctrl+W
Help	Contents	Ctrl+H

The next table includes function key assignments for selecting interaction tools which allow you to change plot view types.

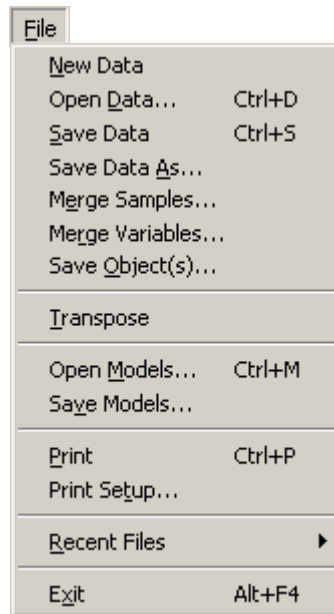
Table 16.2
Tool and View Shortcuts

	To Choose	Press
Interaction Tools	Pointer	F1
	Spinner	F2
	ID	F3
	Zoom	F4
	Range	F5
View Types	Table	F6
	3D Scatter Plot	F7
	2D Scatter Plot	F8
	Multiplot	F9
	Line Plot	F10

File Menu

The File menu provides access to the different actions which deal with manipulating data files and results. This includes creating new files, opening and saving files and models, saving objects and printing.

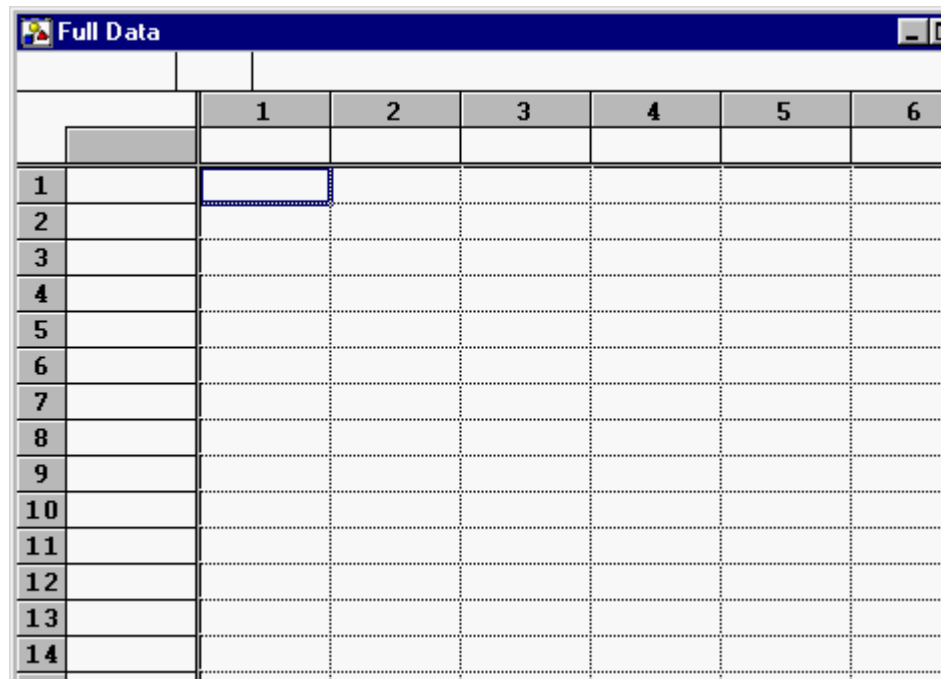
Figure 16.1
File menu



NEW

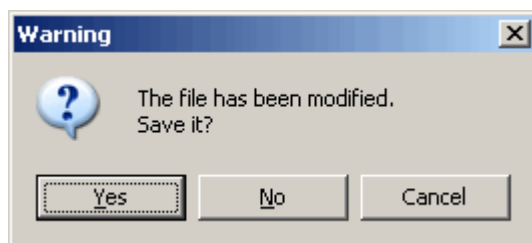
When the New menu item is selected, a blank spreadsheet is displayed as shown below.

Figure 16.2
A new spreadsheet



Selecting New without first saving existing data triggers the following dialog box.

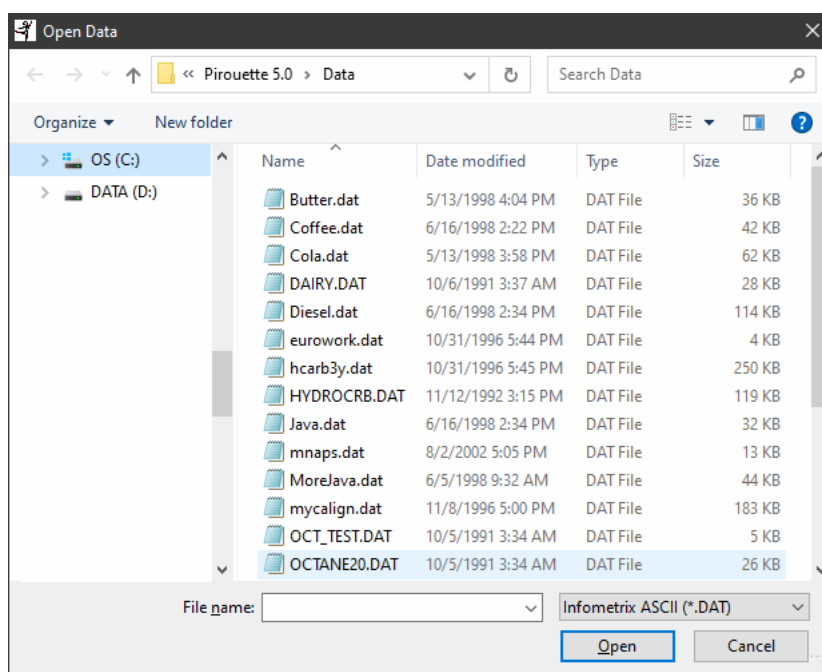
Figure 16.3
Warning dialog for
unsaved results



OPEN DATA

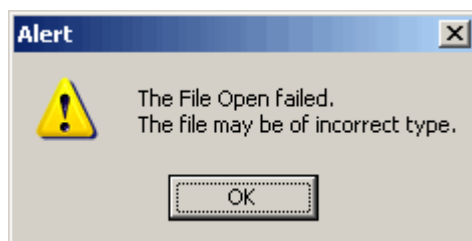
The dialog box displayed when you select the Open Data item in the File menu is shown below.

Figure 16.4
Open Data dialog
box



Changing the directory and drive is accomplished according to standard Windows operations, but the Files of Type field deserves some explanation. Setting this filter determines the file format expected by Pirouette. If the file specified in the File Name box is not in the format specified by Files of Type, the open operation will fail and the message shown below will be displayed.

Figure 16.5
Dialog shown after to
file load fails



If you are uncertain about a file's format, use the *.* filter, which forces Pirouette to try all known formats.

ASCII and Excel files must be in a form understandable by Pirouette. See “ASCII Files” on page 14-5 and “Excel Files” on page 14-9 for more details.

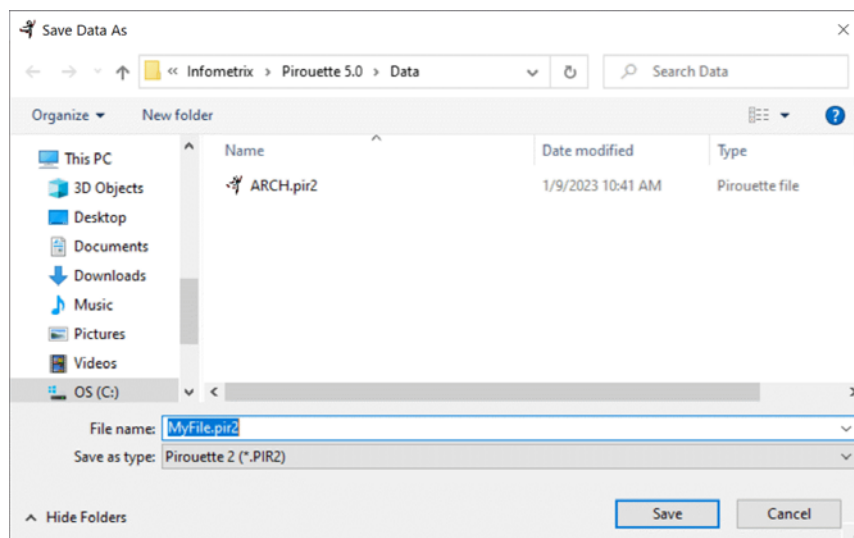
SAVE DATA

Save Data updates the current file with any changes made since the last save. If you originally opened a non-Pirouette file, Save Data opens the Save Data As dialog box discussed in the next section. You can also Save Data with the Save ribbon button.

SAVE DATA AS

Save Data As presents a dialog box which allows you to save your current data and results, if appropriate, to a new file name. You can also change the file type, drive and directory.

Figure 16.6
Save Data As dialog
box



With the Pirouette format, all existing subsets and algorithm results are saved in the file; other formats retain only raw data. To reiterate, saving to a spreadsheet or ASCII file format creates a file which can be opened by other applications but does not contain any subsets or algorithm results generated during the Pirouette session. Only the PIR file type stores this kind of information. See also “Saving Files” on page 15-3.

Note: *Versions of spreadsheet applications (before 2007) allow only 256 columns. The Pirouette format, on the other hand, saves files with a virtually unlimited number of columns. Be aware that if your data has more than 256 columns and you open a saved Excel-format file in an older spreadsheet application, the data may be truncated.*

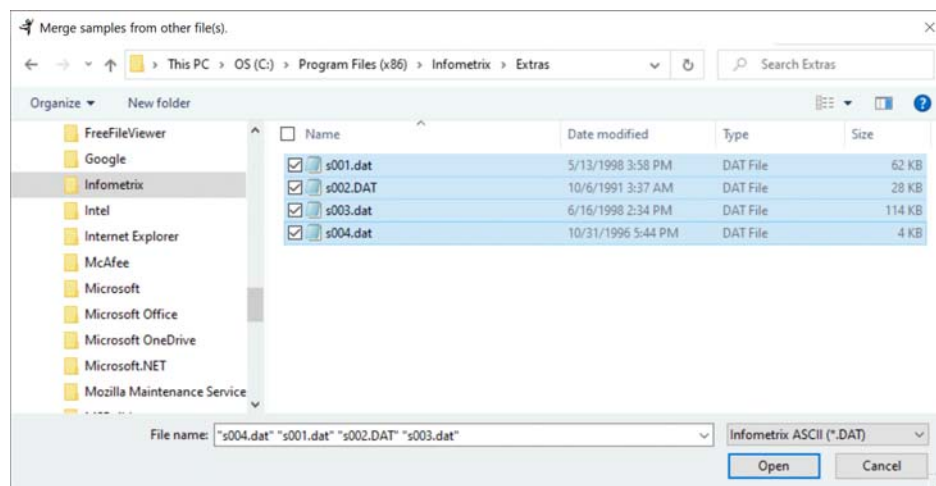
MERGE SAMPLES, MERGE VARIABLES

Pirouette's Merge functions enable you to consolidate a series of files after first opening a file. The Merge functions will present a list box (see, for example, Figure 16.7) where you can select a series of files. All selected files must be in the same subdirectory. To select more than one file, use Shift-click for a contiguous range of files or Ctrl-click for

noncontiguous files. Use Merge Samples to append data (as new samples) to the bottom of the spreadsheet; use Merge Variables to append data (as new variables) to the right of the current spreadsheet.

Note: *When multiple samples are merged, they will be loaded into Pirouette in the order shown in the directory listing of the Merge dialog, whether sorted by Name, Size or Date Modified. The order shown in the File name list is ignored. See also “Merging files from different directories” on page 18-2.*

Figure 16.7
Merge Samples
dialog box



Pirouette does not examine sample or variable names when merging. Thus, when merging samples, if different variable names are present in the original and merged files, the existing column names are retained. Similarly, when merging variables, if different sample names are present in the original and merged files, the existing row names are retained.

The dimensionality of the file(s) need not match, *e.g.*, a merged file can have fewer or more samples or variables than the original file; missing value characters (*) fill the extra cells required to maintain a rectangular data area.

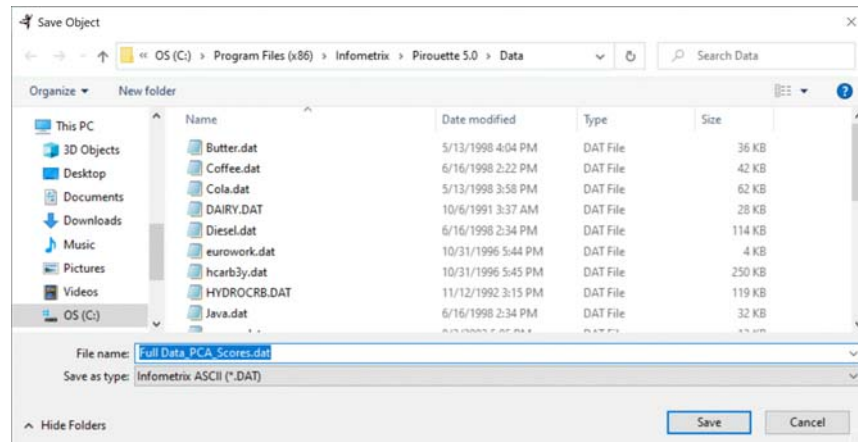
For more information on merging, see “Opening and Merging Existing Data Files” on page 14-3.

Note: *Many spectroscopy data systems store each sample (spectrum) in a single file. Merging many samples into a single file is often the first step in a multivariate investigation.*

SAVE OBJECT(S)

Besides saving a session’s entire results in a Pirouette file, you can also save individual objects in their own files. The Save Objects dialog box is shown below.

Figure 16.8
Save Objects dialog
box



As with Save Data As, you choose the output format with the File Type filter. Save Object works on the active window. In the dialog box, supply an output file name. Note that the suggested name is based on the object's definition and is presented in the File name field. You can keep the suggested name or type an alternate. If the Object Manager window is current, the Save Object(s) menu entry is grayed. See [“Saving Results” on page 15-5](#) for more information.

TRANSDPOSE

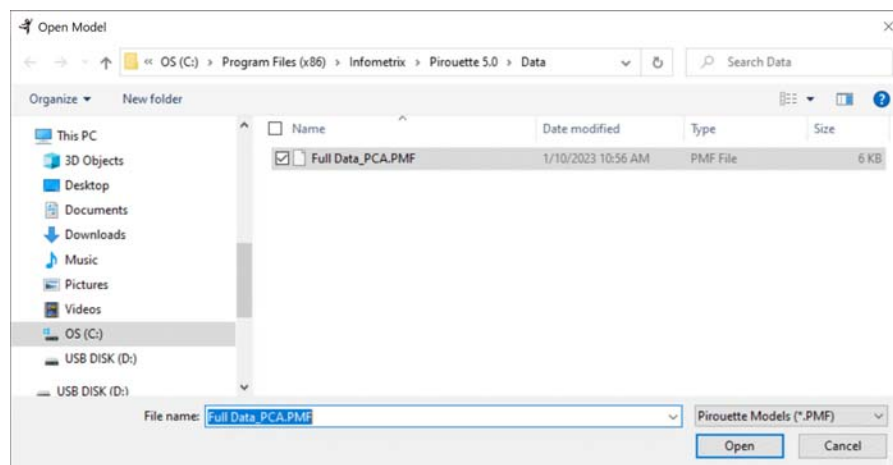
If the data you have loaded into Pirouette has samples in columns, you can easily transpose the data into a more standard row-oriented format with this menu item. See [“Transpose” on page 13-12](#) for more details.

OPEN MODEL

After a modeling algorithm finishes, its associated model is available for predictions. Once saved, this model can be used to make predictions in future Pirouette sessions. The Open Model dialog box is shown below. Only files containing Pirouette-generated models (those with a PMF extension) are displayed.

Note: *With Pirouette 4.0 and earlier, the model format differed and MOD was the default extension for Pirouette models rather than PMF. You may still open some older models with the MOD extension by manually typing the file name with the extension .MOD.*

Figure 16.9
Open Model dialog
box

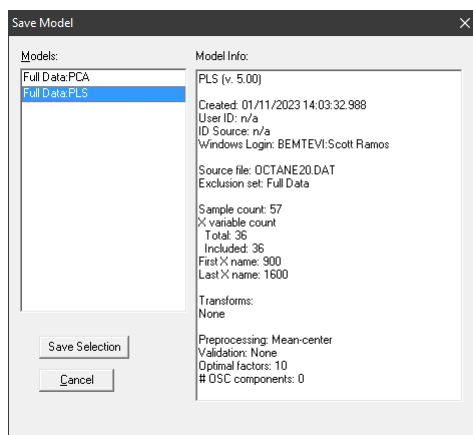


Note: *If you open a model before opening a data file, you will have to reload the model. Similarly, whenever a new file is opened, any previously opened models are lost. Remember to always open the data file first.*

SAVE MODEL

The ability to save models to disk is one of Pirouette's special capabilities. These models can be used in subsequent predictions for any compatible data file. Once a modeling algorithm has been run, its model can be saved via the Save Model dialog box shown in the next figure. A model file can contain models derived from PCA, KNN, SIMCA, CLS, PLS, PCR, PLS-DA, LWR or ALS. All existing models appear in the Models box. Details specific to each are displayed in the Model Info box when a model name is highlighted.

Figure 16.10
Save Model dialog
box



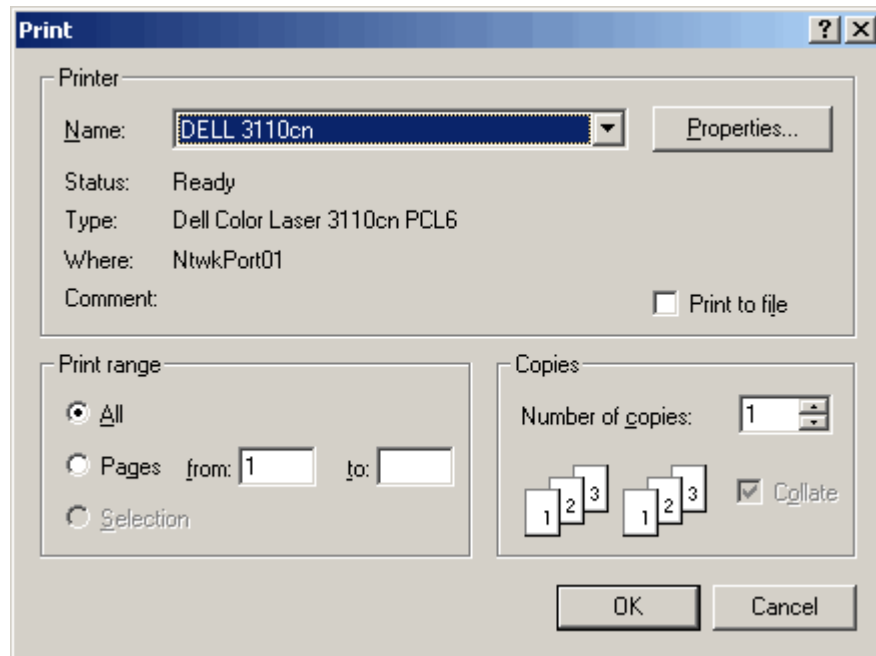
Click on Save Selection and a standard Windows File save dialog will be presented with a suggested model name. Note that PLS and PCR models can also be saved in text format by selecting the ASCII Model Type and in a format used by Galactic's SpectraCalc and GRAMS packages. ASCII models, which cannot be opened by Pirouette, are discussed in "ASCII Models" on page 15-9.

Note: Just as with the Pirouette data file formats, new model formats are made available from time-to-time. Check the web site (see [page 18-14](#)) for the latest information.

PRINT

To print the contents of the current chart window, you will be shown a Print dialog box like that in the accompanying figure. You can select the printer, set the number of copies and/or print to a file.

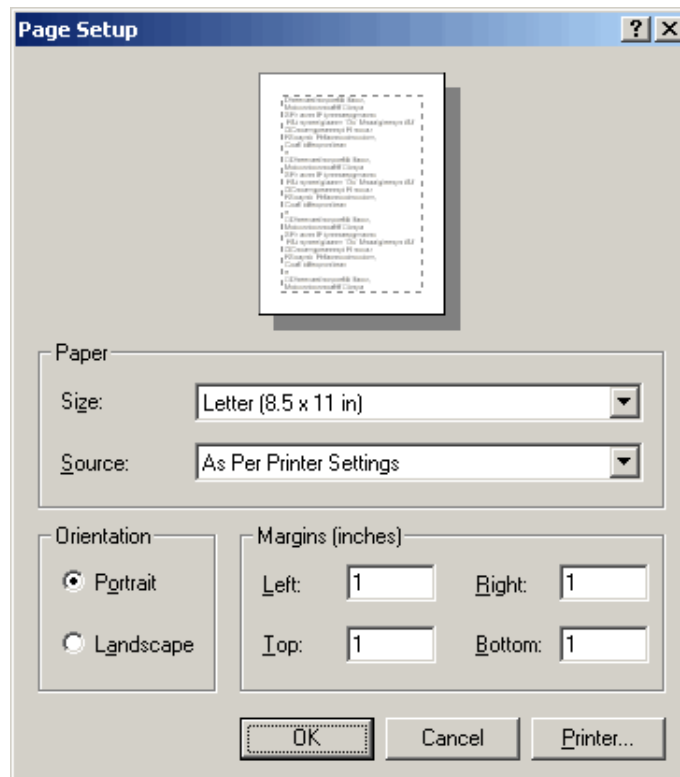
Figure 16.11
Print dialog box



PRINT SETUP

Use the Print Setup dialog box to change the destination printer, paper size and paper tray. For further information on Print Setup, refer to your Windows manual.

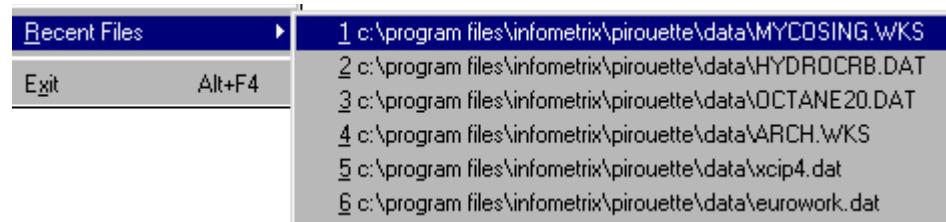
Figure 16.12
Print Setup dialog box



RECENT FILES

Rather than navigating through the Open Data dialog box to find your data file, if it was used recently, its name will appear in the Recent Files list.

Figure 16.13
Recent Files list



EXIT

To quit Pirouette,

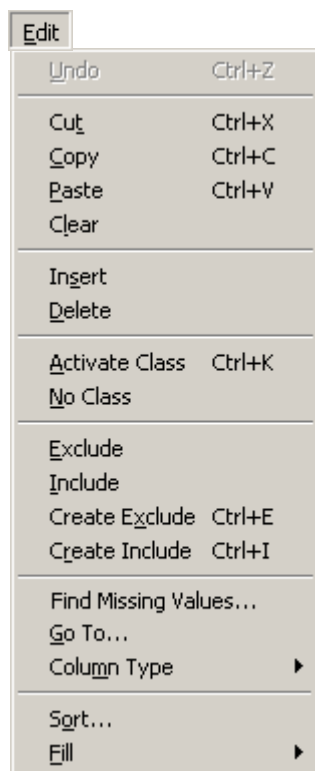
- Select Exit from the File menu or use the Alt-F4 keyboard shortcut.

If you have modified your file since the last save operation, a warning dialog box like that shown in [Figure 16.3](#) is displayed. If you choose Yes and the original format was DAT or format other than PIR, the Save Data dialog box opens. If the original format was PIR, the file is saved with the same name.

Edit Menu

Edit menu items primarily apply to the Pirouette table view of data, although some functions also affect plots. The Edit menu is shown below.

Figure 16.14
Edit menu



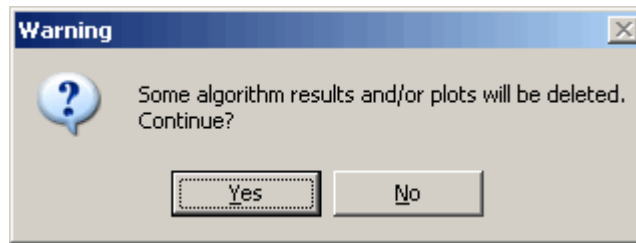
Note: *The Edit menu is context-sensitive; its contents change according to the window which is currently active.*

Because several Edit menu items operate on single or multiple columns and/or rows (as opposed to individual cells), they are grayed unless an entire row or column is selected (*i.e.*, highlighted). Individual or multiple columns or rows can be selected, a particularly useful feature for inserting additional rows or columns or creating an exclusion set. See “[Selecting in Lists and Tables](#)” on page 10-1 for instructions on multiple selections.

Because both columns and rows can be highlighted and because they may be currently off screen, you should clear all highlighting before starting an edit so that you do not inadvertently perform the edit on these hidden cells as well. Clicking once on a cell clears any previous row or column selection. On the other hand, you may want to make a series of discontinuous selections using the Ctrl+click mechanism. In this situation it is possible that there will be selected cells out of the visible range of the table.

An editing action may invalidate existing algorithm results. In this case, a warning like that shown below is displayed. Continuing the edit will cause algorithm results and any related charts to be removed.

Figure 16.15
Warning message
following edits



UNDO

In the Notes window, the information which is initially presented is a brief summary of the parameters set for the algorithm. You may wish to edit this list or add your own comments. After making any additions or deletions, you can change your mind and revert to the text present before your changes were made by using the Undo menu item.

An Undo capability has not been implemented for other actions in Pirouette.

CUT

Cut places selected cells, rows or columns onto the clipboard for later pasting. The cut region is indicated by the so-called marching ants, a box with moving dashed borders. When the paste is effected, the cut region is filled with the missing value indicator (*). If a second cut is performed prior to pasting the contents of the first cut, the values extracted by the second cut operation replaces that from the first cut on the clipboard. To abort a cut (and get rid of those pesky marching ants), press the Esc key before pasting.

To cut a range of data from the spreadsheet,

- Highlight a range of cells or one or more contiguous rows or columns
- Select Cut from the Edit menu

When rows or columns are cut, default names are generated for the cut region after the subsequent paste is completed. If a discontinuous set of rows and/or columns is highlighted (using Ctrl-click), only the last selection is cut. Algorithm results cannot be cut.

COPY

Like Cut, Copy also places selected cells, rows or columns onto the clipboard for later pasting, as well as any text from the Notes window. Use Copy if you need to duplicate data in a different section of the spreadsheet. Like Cut, the marching ants denote the copy region, the subsequent paste applies to the target of the most recent copy, and only the last selection is copied when discontinuous rows and/or columns are highlighted.

To copy a range of data from the spreadsheet,

- Highlight a range of cells or one or more contiguous rows or columns
- Select Copy from the Edit menu

You can also Copy data in result tables to the clipboard for transfer to other applications.

The copy command is also available when a chart window is current. In this case, a bitmap image is transferred to the clipboard. The Copy action does not include the window title bar in the image; however, you may do so by holding the Shift key down when selecting Copy.

In addition to the copying of bitmaps, it is also possible to copy a graphic as a metafile or vector-based image. When a graphic is in the front window, a Copy Special menu item

is enabled in the Edit menu, giving you the choice of saving an Enhanced metafile to either the clipboard or to a file. Saving to a file presents a standard File Save dialog box so that you can name the saved image.

PASTE

When Paste is selected from the Edit menu, the contents of the clipboard are immediately placed in the spreadsheet beginning at the active cell. If you have not previously copied or cut data from the spreadsheet, Paste is disabled. For additional rules governing the size of valid paste areas, see [“Cut, Copy, Paste, and Clear” on page 13-9](#).

Paste is also available for the Notes window. If you have relevant text created in another application, for example, or simply another Notes window inside Pirouette, copy that text to the clipboard, then paste it into the Notes window with which you are working. Upon saving the Pirouette file, that text will be retained as a part of the results from the corresponding algorithm.

CLEAR

Clear immediately removes all values in the selected range. Cleared information is not copied to the clipboard and is therefore irretrievable. Cells are filled with the missing value indicator (*). Cleared row or column labels are not removed but default names are generated.

To clear a range of data,

- Select a range of cells or one or more contiguous rows or columns
- Select the Clear item from the Edit menu

If a discontinuous set of rows and/or columns is highlighted (using Ctrl-click), all selections will be cleared. Algorithm results cannot be cleared.

INSERT

Insert adds new rows or columns to a spreadsheet. The menu item is available only when entire rows or columns are highlighted so clicking on the row or column index is necessary. You cannot insert into algorithm results.

To insert rows or columns,

- Select one or several rows and/or columns
- Select Insert from the Edit menu

New rows and/or columns are inserted just before (*i.e.*, above or to the left of) the rows and/or columns selected. When multiple rows or columns are selected, the same number of rows and/or columns are inserted. The rows and/or columns need not be contiguous. New columns will be of the same variable type as the column to the right. Insert is prevented if columns of different types are selected.

DELETE

Rather than excluding rows or columns (discussed in [“Create Exclude/Exclude” on page 16-15](#)), Delete permanently removes them. The menu item is available only when entire rows or columns are highlighted so clicking on the row or column index is necessary. You cannot delete from algorithm results.

To delete rows or columns,

- Select one or several rows and/or columns

- Select Delete from the Edit menu

The rows and/or columns need not be contiguous.

ACTIVATE CLASS

The ability to activate a class variable provides for:

- Color mapping to sample points and traces, based on their class value
- Indicating categories to classification algorithms
- Automatic creation of new class variables from the HCA dendrogram

When initially displaying a scatter or line plot of samples, the color of points and traces are mapped by row index to the Color Sequence (discussed on [page 10-18](#)). However, if your data includes a class variable, it can provide a color mapping scheme for sample-oriented scatter and line plots. To trigger this mapping,

- Click on the column index of a class variable in a data spreadsheet
- Select Activate Class from the Edit menu

The colors of plotted sample points or traces correspond to their class values but those associated with variables are unaffected.

Another method for activating a class variable uses the Active Class button in the Status Bar. See [page 12-35](#) for details.

A new class variable can be both created and activated from the HCA dendrogram; see “[Creating Class Variables](#)” on [page 12-27](#). A message at the bottom of the Pirouette screen shows name of the active class variable (if any).

NO CLASS

To deactivate a class variable and cancel the associated color mapping,

- Select No Class from the Edit menu, or
- Click the Active Class button and choose None from the list

CREATE EXCLUDE/EXCLUDE

Only included rows and columns are used in the processing by Pirouette’s algorithms. To exclude from a table view and generate a new subset having fewer rows and/or columns,

- Select one or several rows and/or columns
- Select Create Exclude from the Edit menu

The highlighted rows and/or columns change to gray (if that is the current Excluded Background Color set in View Preferences, see [page 10-7](#)) and the title of the spreadsheet becomes *Unnamed-N*. This new exclusion set also appears in the Object Manager. If you Create Exclude from a table view of an algorithm result, the result window is unchanged but the Object Manager is updated to show that a new subset has been created.

To exclude additional rows and/or columns but accumulate these changes without generating a new subset,

- Select one or several rows and/or columns
- Select Exclude from the Edit menu

It is sometimes more convenient to perform exclusions from a graphic than from the spreadsheet. The Create Exclude item is available when points are selected in dendrograms and scatter plots of both data and results. However, you cannot choose the Exclude

item for results graphics (this would invalidate the algorithm results). For more information, see “Creating Subsets from a Graphic” on page 12-32. For a general discussion of subset creation, see “Creating Subsets from Tables” on page 13-20.

CREATE INCLUDE/INCLUDE

To include from a table view of data and generate a new subset,

- Select one or several previously excluded rows and/or columns
- Select Create Include from the Edit menu

The highlighted rows and/or columns will no longer be gray, the title of the spreadsheet window changes to *Unnamed-N* and the Object Manager gets a new set entry.

To include additional rows and/or columns but accumulate these changes without generating a new subset,

- Select one or several previously excluded rows and/or columns
- Select Include from the Edit menu

Note: To create a duplicate of the current subset from its table view, highlight an already included row or column and then select Include.

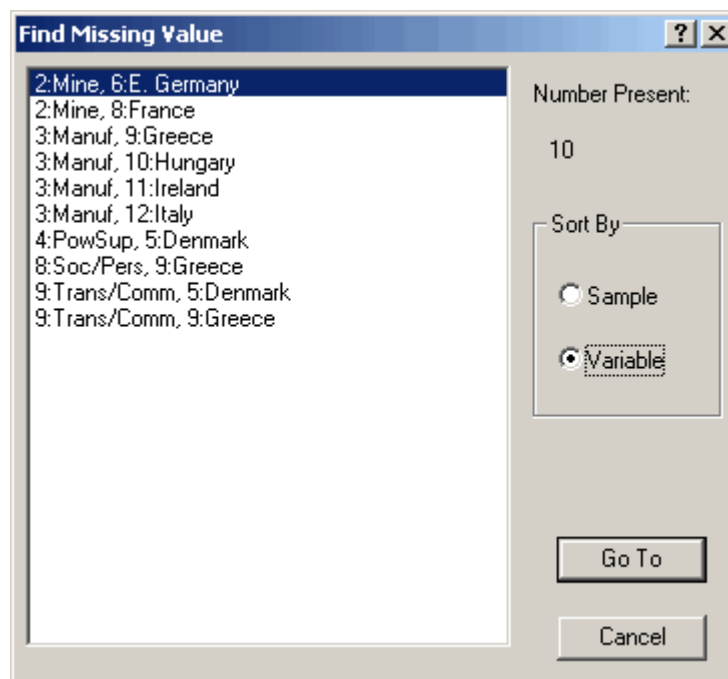
FIND MISSING VALUES

Data sets may be missing values because some data were simply not collected or because a merge operation created ranges of missing value to keep the matrix rectangular. If it is not obvious where the missing values fall, this utility can help you locate them.

- Select Find Missing Values from the Edit menu

This action will present the following dialog box.

Figure 16.16
Find Missing Values
dialog



You can sort the list either in order of variables (the default view) or by sample. Click on an item in the list, then on the Go To button to return to the spreadsheet, and the selected row and column will be positioned in the upper left of the table.

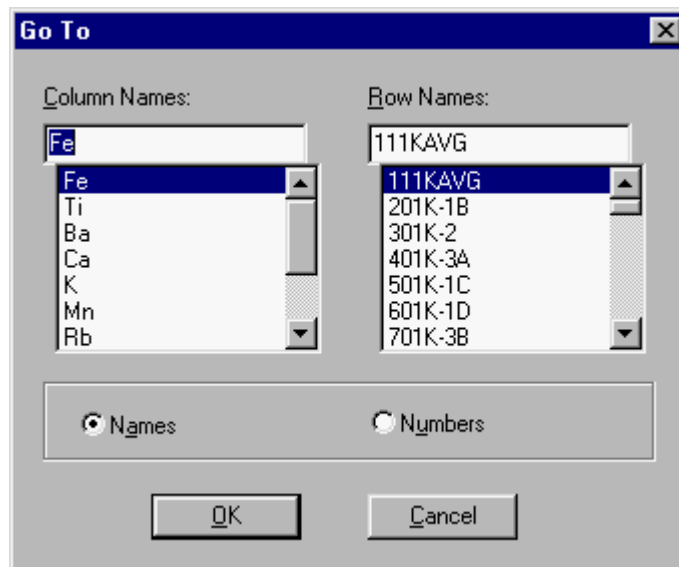
Go To

It is sometimes awkward to scroll to a desired location in a large table, particularly in data sets with large numbers of samples and/or variables. To move to a specific cell immediately,

- Select Go To from the Edit menu

The dialog box which is shown provides several ways to specify the destination.

Figure 16.17
Go To dialog box



The most common approach is:

- Scroll through either list until the row/column name or number appears
- Click on its name

If you know the row and/or column name,

- Type in the full Row or Column name (not case sensitive)

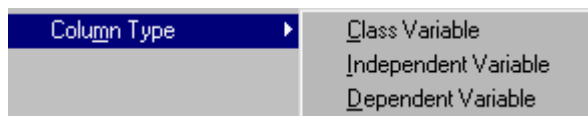
The Selection Type radio button switches between lists of Names and Numbers (*i.e.*, column or row indices). After you click on OK, the spreadsheet is redisplayed such that the designated cell is at the top-left corner of the window.

Although data for class and dependent variables are in a different portion of the spreadsheet, Go To operates in the same manner. The only difference is that when Numbers are selected, you must type a C or a Y first to access the corresponding column numbers.

COLUMN TYPE

Pirouette variables can be of three different types: independent, dependent, or class. To assign a different type, highlight that column, and select Column Type, which opens the submenu shown below. Select the new type for the column(s) you have highlighted.

Figure 16.18
Column Type
submenu



The column is moved to the end of the variable block region of the type chosen from the submenu. For further discussion on the Column Type command, see [“Changing Variable Types”](#) on page 13-10.

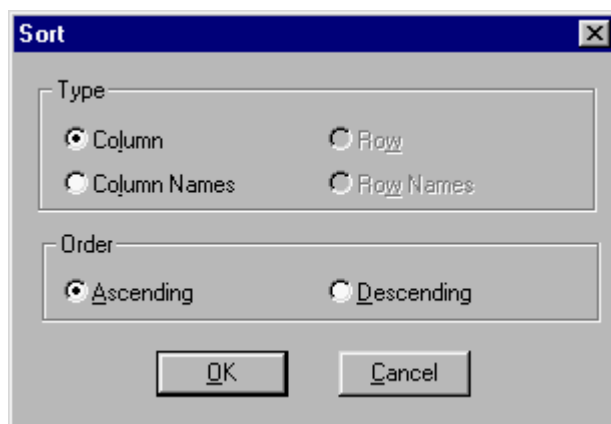
Note: *Class values must be integers. Whenever you change an X or Y variable to a class variable, a warning is put up reminding you that the values in the column will be truncated.*

SORT

To order your data,

- Highlight at least two rows or columns to be sorted
- Select Sort from the Edit menu

Figure 16.19
Sort dialog box

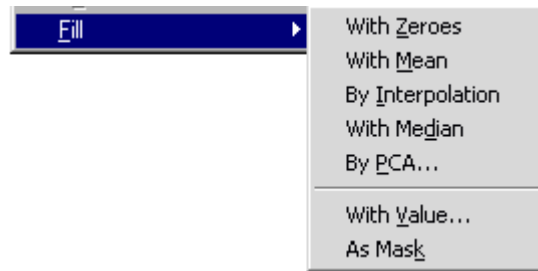


The dialog box allows you to sort ascending or descending and by value or name. The sort key is that row or column which contains the active cell. See [“Sorting Data”](#) on page 13-11 for a sort example.

FILL

Some data sets may have missing values (see also [“Find Missing Values”](#) on page 16-16). With Pirouette you can fill missing values in several ways. Fill works on single or multiple columns or rows. First highlight the applicable region, then select Fill which opens the submenu shown in [Figure 16.20](#).

Figure 16.20
Fill submenu



The Fill submenu lists seven options:

With Zeros Missing values in the highlighted region are filled with zero.

With Mean Missing values in the highlighted region are filled with column means (if columns are selected) or with row means (if rows are selected).

By Interpolation Missing values in the highlighted region are filled with an interpolated value. When a row is selected, row values on either side of the missing value are used. When a column is selected, column values above and below the missing value are used. A linear interpolation is applied in the case of multiple contiguous missing values. If the missing value is at the edge, the first (last) value replaces the missing value(s).

With Median Missing values in the highlighted region are filled with column medians (if columns are selected) or with row medians (if rows are selected).

By PCA Iterative PCA modeling and prediction of the missing values is run until convergence when the final predictions are retained to substitute the missing values. If a sub-range of columns and/or rows is selected, the algorithm will be run on only those values.

With Value Missing values can be filled with a specific value. Only the values at the intersection of simultaneously highlighted rows and columns are filled.

As Mask Missing values in a highlighted row are filled with either a one or zero. Ones are inserted for highlighted columns, otherwise zeros are inserted.

For more information, see [“Filling Missing Values” on page 13-13](#).

NEW SET

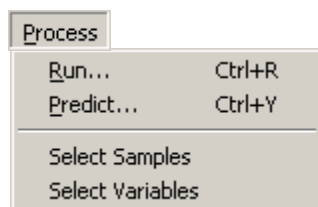
When the Object Manager is the frontmost window, an additional item is enabled in the Edit menu, called New Set. Selecting this option will create, and display, a new subset in which all rows and columns are included. It performs the same function as drag and drop of the Disk icon in the Object Manager (see [“Subsets” on page 11-9](#)).

Process Menu

This discussion focuses on the steps needed to run Pirouette algorithms. For detailed descriptions of the algorithms, refer to the appropriate chapters in Part II [Guide to Multivariate Analysis](#).

The Process menu is the heart of Pirouette; from there algorithms and predictions are configured and initiated.

Figure 16.21
Process menu



RUN

Running algorithms is accomplished via the Run item on the Process menu which opens the Run Configure dialog box shown in [Figure 16.22](#). To configure an analysis,

1. Click on an entry in the Exclusion Set box
2. Click on an entry in the Algorithm box
3. Choose any needed Transforms
4. Click on Run

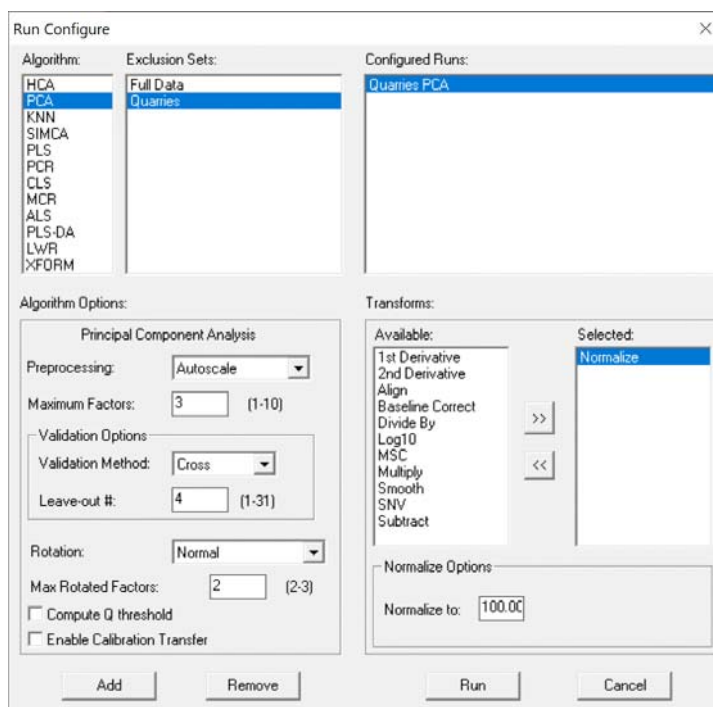
To configure more than one analysis, repeat steps 1 - 3, then

- Click on Add

Continue adding subset-algorithm pairs to the list as needed. When your batch list is complete and you want to begin executing algorithms,

- Click on Run

Figure 16.22
Run Configure
dialog box



Algorithm Options

Each algorithm has default options which you may want to change. The following discussion lists those options but does not discuss why one setting is preferred over another. Each preprocessing setting is defined in [“Preprocessing” on page 4-26](#). Algorithm-ori-

ented chapters in Part II [Guide to Multivariate Analysis](#) also offer advice on algorithm options.

To establish an algorithm configuration,

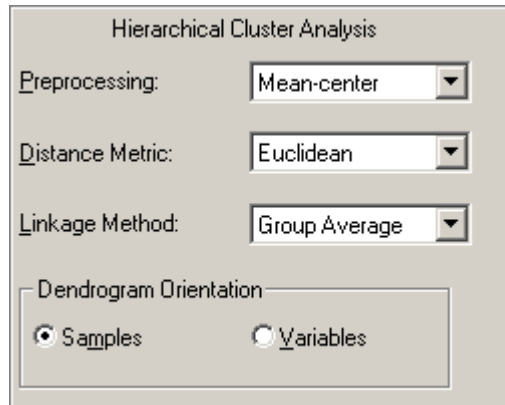
- Highlight the Algorithm
- Change the various settings shown in the Algorithm Options box
- Select and configure zero, one or more Transforms
- Highlight an Exclusion set
- Click on Add or Run

After a particular algorithm’s options are changed, the settings stay in effect until you change them again. However, if you change the dimensionality of the data set by deleting or excluding samples or variables, valid ranges for options may be impacted.

HCA

The HCA options are shown below. When Run Configure is first opened, HCA is selected by default.

Figure 16.23
HCA options



Choices for the HCA options are listed in the following table. See [“Linkage Method Definitions” on page 5-2](#) for a detailed discussion of this topic.

Table 16.3
HCA Choices

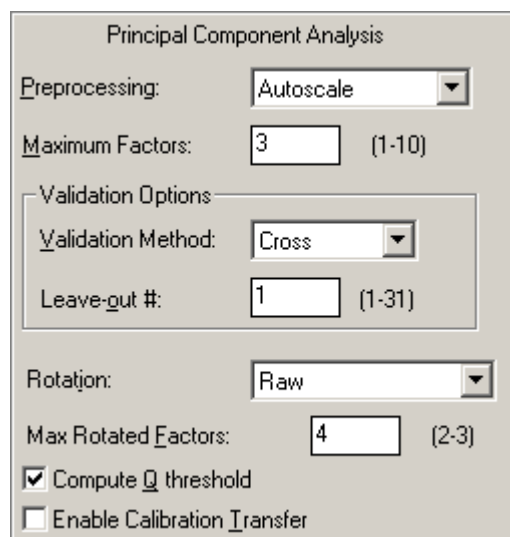
Option	Default	Choices
Preprocessing	None	None Autoscale Mean Center Variance Scale Range Scale Pareto
Distance Metric	Euclidean	Euclidean Euclidean (no init)
Linkage Method	Single	Single Centroid Complete Incremental

Option	Default	Choices
		Median Group Average Flexible
Dendrogram Orientation	Samples	Samples Variables

PCA Options

The PCA options are shown in the next figure.

Figure 16.24
PCA options



Choices for the PCA options are listed below. See “[Varimax Rotation](#)” on page 5-28 for a detailed discussion of the rotation choices. For a discussion of the Validation Method and Leave-out # choices, see “[Validation-Based Criteria](#)” on page 7-6. Computation of the threshold for the Q statistic will be computed only if the option is checked.

Table 16.4
PCA Choices

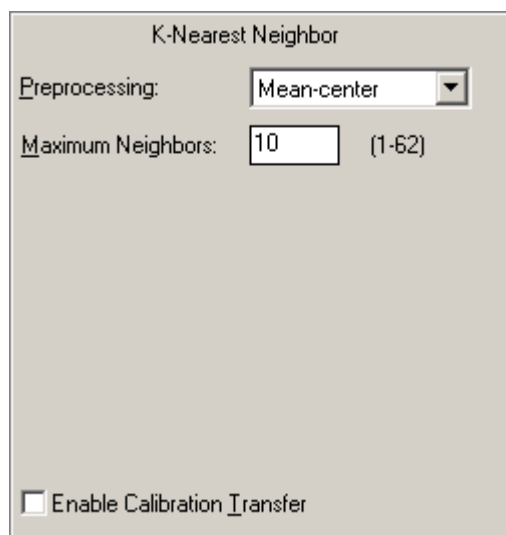
Option	Default	Choices
Preprocessing	None	None Autoscale Mean Center Variance Scale Range Scale Pareto
Rotation	None	None Raw Normal Weighted Weighted-Normal
Maximum Factors	Varies, up to 10	Varies, up to lesser of # of samples and variables

Option	Default	Choices
Max # Rotated Factors	Varies, up to 10	Varies, up to lesser of # of samples and variables
Validation Method	None	None Cross Step
Leave-out #	1	1 up to one half the number of samples
Compute Q threshold	Off	Off, On
Enable Calibration Transfer	Off	Off, On

KNN Options

The KNN options are shown in the next figure.

Figure 16.25
KNN options



Choices for the KNN options are listed in the following table. For a detailed discussion of the algorithm, see [“Mathematical Background”](#) on page 6-3.

Table 16.5
KNN Choices

Option	Default	Choices
Preprocessing	None	None Autoscale Mean Center Variance Scale Range Scale Pareto
Maximum Neighbors	Varies, up to 10	Varies, up to lesser of # of samples and variables
Enable Calibration Transfer	Off	Off, On

SIMCA Options

The SIMCA options are shown below.

Figure 16.26
SIMCA options

Choices for the SIMCA options are listed in the table below. For a detailed discussion of the algorithm, see [“Mathematical Background”](#) on page 6-16.

Table 16.6
SIMCA Choices

Option	Default	Choices
Preprocessing	None	None Autoscale Mean Center Variance Scale Range Scale Pareto
Scope	Local	Local Global
Maximum Factors	Varies	Varies (determined by class with minimum number of variables/samples)
Probability Threshold	0.95	0.01 to 0.9999
Enable Calibration Transfer	Off	Off, On

PCR and PLS Options

The PCR and PLS options are identical and are shown in the next figure.

Figure 16.27
PCR (and PLS)
options

Choices for the two algorithms are listed in the table below. For a detailed discussion of both algorithms, see [“Mathematical Background”](#) on page 7-3. See [“Model Validation”](#) on page 5-19 for a discussion of the Validation Method and Leave-out # choices.

Table 16.7
PCR and PLS
Choices

Option	Default	Choices
Preprocessing	None	None Autoscale Mean Center Variance Scale Range Scale Pareto
Validation Method	None	None Cross Step Active Class
Leave-out #	1	1 up to one half the number of samples
Maximum Factors	Varies, up to 10	Varies, up to lesser of # of samples and variables
# OSC Components	0	0 - 3
Enable Calibration Transfer	Off	Off, On

CLS options

Options for CLS differ from those of PCR and PLS because model optimization is based not on the number of factors, rather on the form of the baseline.

Figure 16.28
CLS options

Choices for the CLS algorithm are listed in the following table. For a detailed discussion of the algorithms, see [“Mathematical Background” on page 7-44](#). See [“Model Validation” on page 5-19](#) for a discussion of the Validation Method and Leave-out # choices.

Table 16.8
CLS Choices

Option	Default	Choices
Probability Threshold	0.95	0.01 to 0.9999
Validation Method	None	None Cross Step
Leave-out #	1	1 up to one half the number of samples
Enable Calibration Transfer	Off	Off, On

ALS options

Most of the ALS options concern the constraints applied during the least squares optimization.

Figure 16.29
ALS options

Choices for the various ALS options are shown in the following table.

Table 16.9
ALS Choices

Option	Default	Choices
Preprocessing	None	None Autoscale Mean Center Variance Scale Range Scale Pareto
Maximum #of sources	Varies, up to 10	Varies, up to lesser of # of samples and variables
Non-negativity	off	off, on
Unimodality	off	off, on
Closure	None	None, Amounts, Profiles
Initial estimates, from	Rows	Rows, Columns

PLS-DA Options

Options for the PLS-DA algorithm are a blend of those from PLS and the classification algorithms.

Figure 16.30
PLS-DA options

Choices for the various options available to PLS-DA are tabulated below.

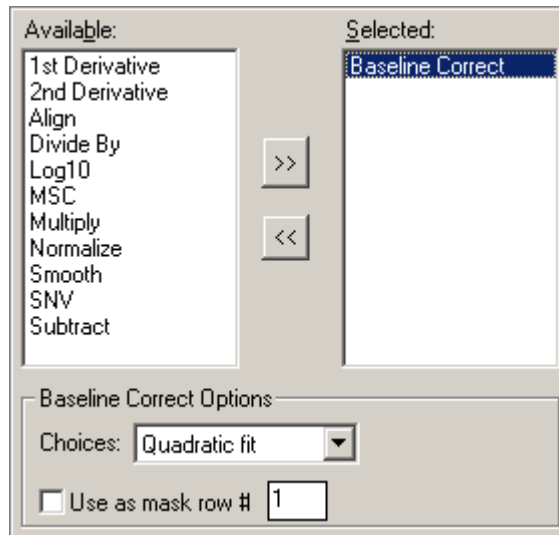
Table 16.10
PLS-DA choices

Option	Default	Choices
Preprocessing	None	None Autoscale Mean Center Variance Scale Range Scale Pareto
Validation Method	None	None Cross Step
Leave-out #	1	1 up to one half the number of samples
Maximum Factors	Varies, up to 10	Varies, up to lesser of # of samples and variables
Class Variable	First category in table	List of all available categories
# OSC Components	0	0 to 3

Transforms

The Transform dialog is shown in the next figure. Any transform or combination of transforms appearing in the Selected box when the Add or Run button is clicked will be applied to the independent variables before preprocessing and algorithm execution.

Figure 16.31
Transform options



To select a transform,

- Highlight the desired transform in the Available list
- Click on >>

De-selecting a transformation works the same way:

- Highlight the desired transform in the Selected list
- Click on <<

You can also double-click on the transform name to move it to the opposite list. Any number of transformations may be selected, in any order, although an individual transform cannot be selected twice. Transforms are applied in the order selected. Some transforms can be customized. Possible choices are shown in [Table 16.11](#); however, it is not necessary to use the options (such as Use mask). For more information about each choice, see [“Transforms” on page 4-10](#).

Table 16.11
Transform choices

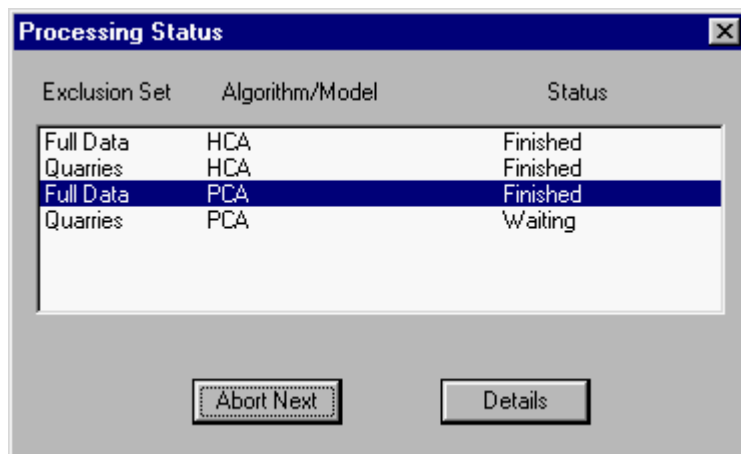
Transform	Option	Default	Choices
Derivative	# of Points	5	5 to 95 (pop-up list)
Smooth	# of Points	5	5 to 95 (pop-up list)
Log 10			none
Multiply	Factor	1.000	
Normalize	Factor	100.00	
Subtract	Value	0.000	
	at Var #	1	any excluded variable
Baseline correct	Subtract sample	1	row number
	Linear fit		use mask
	Quadratic fit		use mask
	Cubic fit		use mask
Divide by	Sample 2-norm		use mask
	Sample 1-norm		use mask
	Sample max		use mask
	Sample range		use mask
	at Var #	1	variable number
	Sample vector	1	row number
	Subset mean		
MSC			use mask
SNV			none
Align	Window size	5	0, or 5 to 0.5 xVariablesTotal
	Align to row #	1	row number

Note: To see the effect of a transform on the independent variables, run the XFORM algorithm with the transform(s) of interest. If you prefer to work only with transformed data, you can save the transformed result using Save Objects and then later merge any class and dependent variables.

Run Status

After a batch of analyses has been set to processing, a dialog box is presented with a list of all the processing to be done.

Figure 16.32
The Run Status
dialog box



If a process cannot complete, it will abort and the next process will start. When all the analyses are finished, the dialog will disappear unless there was an aborted process. You can determine the cause of an abort by double-clicking on the abort message or clicking on the Details button. For a list of possible causes for aborts, see [“Processing Errors” on page 18-5](#).

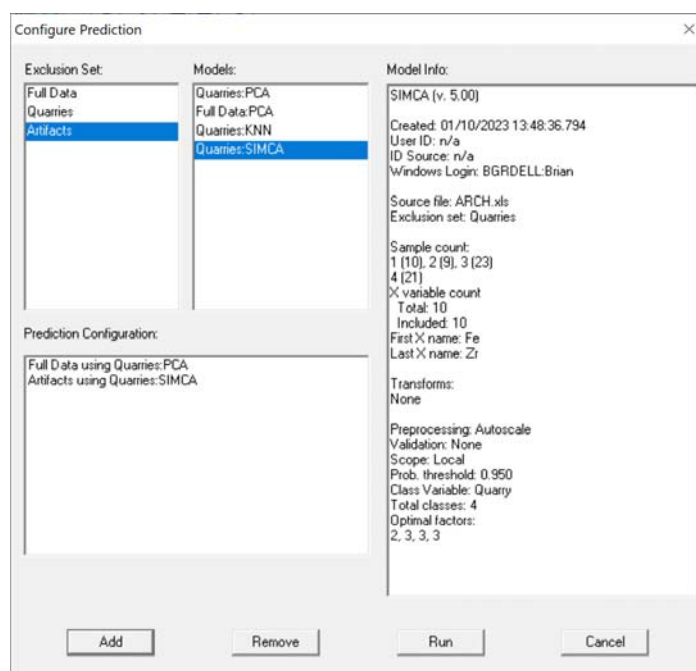
If you have chosen to batch several processes at once, but then decide to terminate the batch before it is completed, you can force an abort by clicking on the Abort Next button in the dialog box. The processing will stop after the current process is complete, and the remaining processes will not run.

By default, no results are shown after processing is complete. Evidence of completion is the existence of folders of results shown in the Object Manager. You can ask Pirouette to display results on completion via a preference (see [“Number of Plot Windows” on page 10-17](#)), but keep in mind that when many algorithms are in a batch, it can take a noticeable amount of time, with large data sets, for all plots to be presented. After all processing is complete, you can still display results for any process by dragging the appropriate icons from the Object Manager.

PREDICT

A multivariate model is created whenever PCA, KNN, SIMCA, CLS, PCR, PLS, PLS-DA, LWR or ALS is run. Making predictions with these models is accomplished via the Predict item on the Process menu which opens the Predict Configure dialog box shown in next.

Figure 16.33
Predict Configure
dialog box



To configure a prediction,

- Highlight an Exclusion Set
- Highlight a Model
- Click on Run (or Add to make batch of predictions)

Note: *If there are variable exclusions in a subset, its name will be greyed in the Exclusion Set list. To be available for predictions a subset must have all variables included.*

If the total number of variables in the subset does not match the number of variables used in the training set from which the model was created, Predict will not run, and an Alert message will be displayed.

When you highlight a model, information about it is displayed in the Model Info box. For more explanation on how to create a model and run a prediction, refer to the respective chapter (for example, see [“Making a SIMCA Prediction”](#) on page 6-26 and [“Making a PCR/PLS Prediction”](#) on page 7-31).

Note: *Support for very old models (version 2.03, 1997 and older) was discontinued. You may load the model, but when you click on Model Info in the Configure Prediction dialog, this message is shown: “This model version (2.03) is no longer supported.”*

SELECT SAMPLES

Selecting this menu item brings up the Sample Select dialog.

Figure 16.34
Sample Select dialog

The options available for performing sample selection are described in the following table.

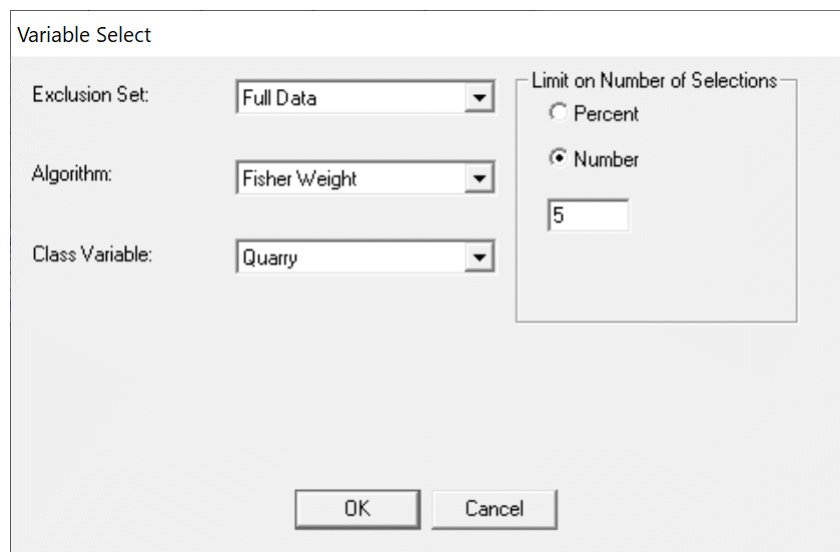
Table 16.12
Sample selection options

Parameter	Default	Choices
Exclusion Set	Full Data	Any subset
Algorithm	[persistent]	Kennard-Stone Orthogonal Leverage PCA Hypergrid Random
Class Variable	None	Any class variable
Create complement set	Off	Off, On (Random only)
Limit on Number of Selections	[persistent]	Percent, 1 to 99 Number, 1 to 1 less than total number of samples

SELECT VARIABLES

Selecting this menu item brings up the Variable Select dialog.

Figure 16.35
Variable Select dialog



The options available for performing variable selection are described in the following table.

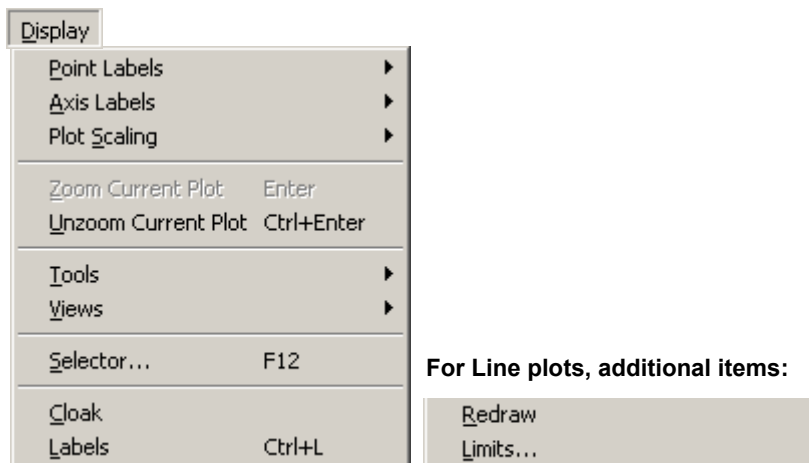
Table 16.13
Variable selection options

Parameter	Default	Choices
Exclusion Set	Full Data	Any subset
Algorithm	[persistent]	Standard Deviation Rank Fisher Weight Variance Weight
Class Variable* [* not available for STR; required for FW, VW]	None	Any class variable
Limit on Number of Selections	[persistent]	Percent, 1 to 99 Number, 1 to 1 less than total number of variables

Display Menu

The Display menu, shown below, provides Pirouette users with tools for modifying the appearance of charts.

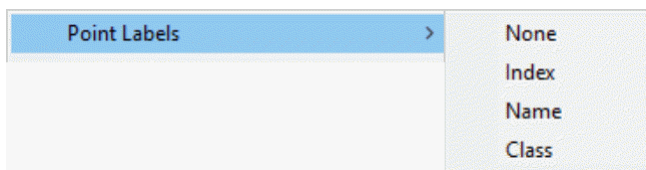
Figure 16.36
Display menu items



POINT LABELS

Pirouette allows you to label the points of the current 2D or 3D plot with either a number (the row index) or name (the sample label). When the Point Labels item is selected, a submenu with three options is deployed.

Figure 16.37
Point Labels submenu



- **None** removes labels from the plot
- **Index** labels each point with its index number
- **Name** labels each point with a name
- **Class** labels each point with its Class value

The number or name displayed depends on the object plotted. Typically, it corresponds to a row/column number or name. The current label type is grayed.

AXIS LABELS

When the Axis Labels item is selected, a submenu with two options is deployed.

Figure 16.38
Axis Label submenu

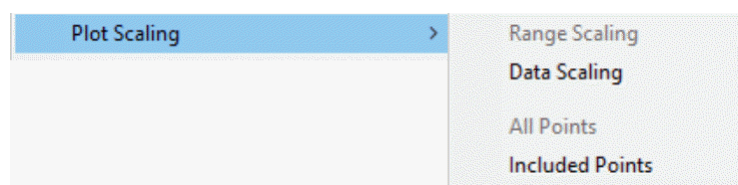


- **Number** labels the axes with row or column indices or numbers
- **Name** labels the axes with row or column names (e.g., discrete names or wavelength values)

PLOT SCALING

When the Plot Scaling item is selected, a submenu with four options is deployed.

Figure 16.39
Plot Scaling
submenu



- **Range Scaling** plots data scaled to the range on each axis
- **Data Scaling** plots data scaled to the maximum axis range (for the currently displayed axes)
- **All Points** bases the range calculation on all points in the data set
- **Included Points** bases the range calculation only on points included in the subset

ZOOM CURRENT PLOT

To view a subplot so that it fills the window, use one of the techniques listed in [Table 12.4, “Zooming and Unzooming,” on page 12-20](#), which includes the Zoom Current Plot item in the Display menu.

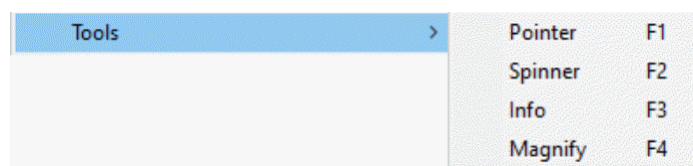
UNZOOM CURRENT PLOT

To shrink a plot to its subplot array format, use one of the techniques listed in [Table 12.4, “Zooming and Unzooming,” on page 12-20](#), which includes the Unzoom Current Plot item in the Display menu.

TOOLS

When a graphical view of data or results is displayed, several interaction tools are enabled. Choosing an item from the Tools submenu is equivalent to clicking on the corresponding ribbon button. The function of each tool is described briefly below.

Figure 16.40
Tools submenu

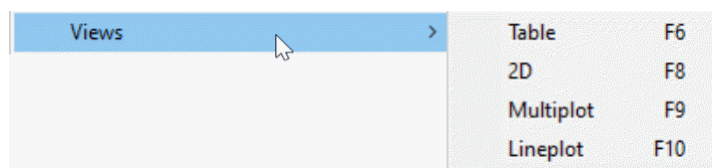


- **Pointer Tool:** selects points in a dendrogram or scatter plot (see [“Selecting Points” on page 12-5](#)) or selects lines in a line plot (see [“Selecting Lines” on page 12-17](#))
- **Spinner Tool:** rotates 3D plots in any direction (see [“Spinning a 3D Plot” on page 12-9](#))
- **ID Tool:** displays row/column number and name for the nearest point or trace (see [“Identifying Points” on page 12-7](#) and [“Identifying Lines” on page 12-15](#))
- **Magnify Tool:** magnifies a portion of a line or scatter plot (see [“Magnifying Regions” on page 12-8](#))
- **Range Tool:** allows graphical selection of variables in line plots (see [“Selecting Ranges” on page 12-18](#))

VIEWS

You can change the plot type of the current chart using either a ribbon button (listed in [Table 10.5, “Ribbon buttons for view switching,”](#) on page 10-4) or an item in the Views submenu.

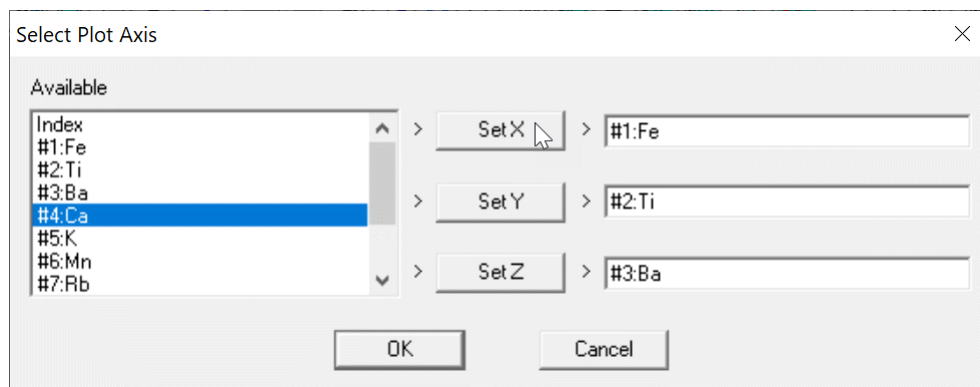
Figure 16.41
Views submenu



SELECTOR

This item allows you to specify which items will show in a plot; a Selector ribbon button is also available (see [“Ribbon Buttons”](#) on page 10-3). For a discussion of each type of Selector dialog see [page 12-5](#) for 2D and 3D scatter plots, [page 12-14](#) for line plots, and [page 12-21](#) for multiplots.

Figure 16.42
Selector dialog box
for 3D plots



CLOAK

The Cloak item is shown when 2D or 3D scatter plot or line plot views are active; it is equivalent to the corresponding ribbon button discussed in [“Cloaking”](#) on page 12-8.

REDRAW

Redraw is shown in the Display menu, only for Line plots. Its action is equivalent to clicking on the Redrawn button in the ribbon, discussed in [“Redrawing Traces”](#) on page 12-19.

LIMITS

Line plots can be zoomed graphically or by selecting this menu item. See [“Magnifying Regions”](#) on page 12-15 for details.

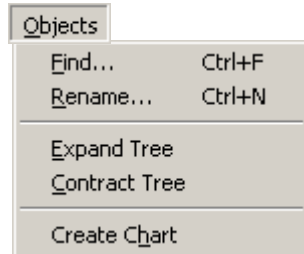
LABELS

The Labels item toggles the visibility of labels and is shown when 2D or 3D scatter plot views are active; it is equivalent to the corresponding ribbon button discussed in [“Point Labels”](#) on page 12-7.

Objects Menu

Most of your interaction with the Object Manager is via the mouse; however, a few functions are available from the Objects menu, shown below.

Figure 16.43
Objects menu



Find and Rename must be accessed via the menu or the corresponding shortcut. The remaining three items can be invoked via a ribbon button or mouse action.

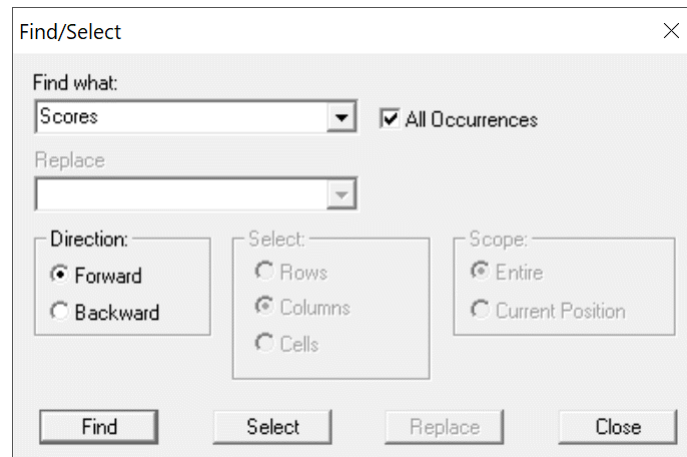
FIND

When only one or two algorithms have been run, it is easy to locate their results in the Object Manager. However, if you have run many algorithms on a variety of subsets, it may be difficult to find a particular Object Manager entity.

To locate a specific object,

- Select Find from the Objects menu
- Type a (case sensitive) object name in the box under Find What
- Click on Find

Figure 16.44
Find dialog box



To highlight all objects whose names contain the given text string,

- Type in the text string
- Check the All Occurrences box
- Click on Select

You can use the Find operation to locate all scores plots, for example, to make a set of plots at once for intercomparisons. The object must be visible in the Object Manager to be selectable.

RENAME

You can rename subsets and folders containing algorithm results; all other object names are fixed and not editable. To change a subset or algorithm results folder name,

- Highlight the object in the Object Manager
- Select Rename from the Objects menu

The following figure shows the dialog box which will open. Type in a new name and click on OK. The new name will appear in the Object Manager and in the title of related charts.

Figure 16.45
Rename dialog box



EXPAND TREE/CONTRACT TREE

Use these menu items to grow or shrink the Object Manger trees; they duplicate the function of the ribbon buttons shown in [Table 10.7, “Ribbon buttons for navigation aids,” on page 10-5](#).

Normally, the trees expand and contract by one level for every request. However, if a tree was fully expanded, and you have closed it entirely by double-clicking on the data file icon, the Object Manager remembers its previous position and the next expand request opens the tree to that level.

CREATE CHART

Dragging and dropping from the Object Manager is a quick and easy way to create new charts. To create new charts from the menu,

- Highlight an item (or several) in the Object Manager
- Select Create Chart from the Objects menu

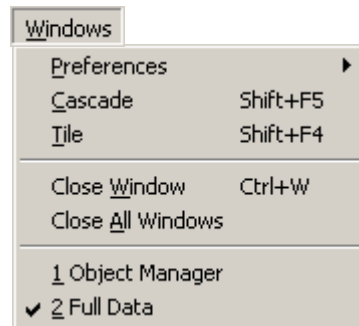
If several items are selected, dragging and dropping them as a group will create a new User window with all of the objects arrayed as subplots. If you hold the Shift key down while dropping, each object will be displayed in its own window.

Windows Menu

In addition to some standard Windows entries, this menu allows you to customize Pirouette through a series of preferences. The last items in this menu (those after Close All

Windows) are the titles of all open windows; the active window is denoted by a check. To make a different window active and bring it to the front (if it is behind another window), select it from the Windows menu.

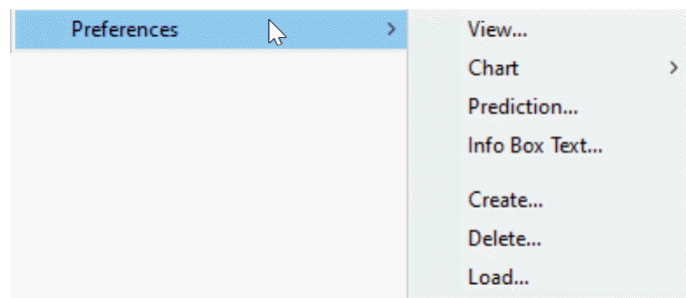
Figure 16.46
Windows menu



PREFERENCES

Pirouette ships with many default settings. Although we have tried to choose generally appropriate values, you may wish to override them via the Preferences item in the Windows menu. Preferences are grouped into four categories briefly described below.

Figure 16.47
Preferences submenu



View

Most graphical and tabular display components can be customized, including colors and fonts. After selecting the View submenu item which opens the dialog box shown below, you can specify the view and attribute to customize. Clicking on a View and double-clicking on an Attribute opens another dialog box where the changes are made. For more details on the changes possible for each view, see [“View Preferences” on page 10-7](#).

Figure 16.48
3D View Preferences
attributes

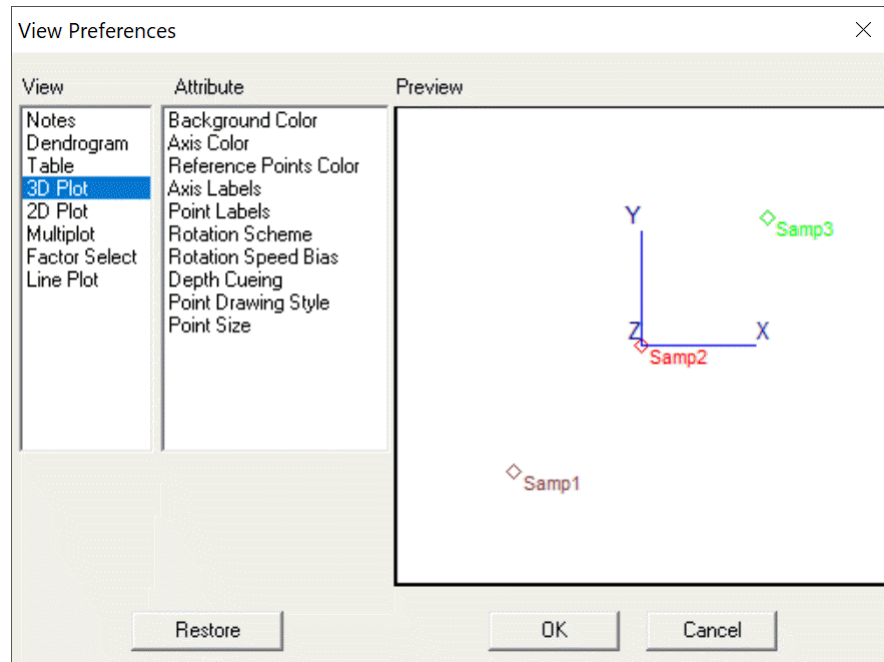


Chart - Label Attributes

Labels which are shown by default on new plots are governed by the settings in the Label Attributes dialog box. See [“Label Attributes” on page 10-16](#) for details.

Figure 16.49
Plot Label Attributes
dialog

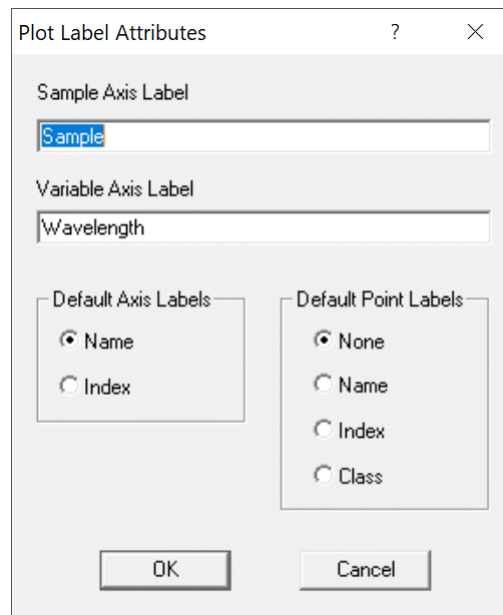


Chart - Window Attributes

Window behavior, including default size and position, are governed by the Window Attributes dialog. See [“Window Attributes” on page 10-17](#) for explanations.

Figure 16.50
Plot Window
Attributes dialog

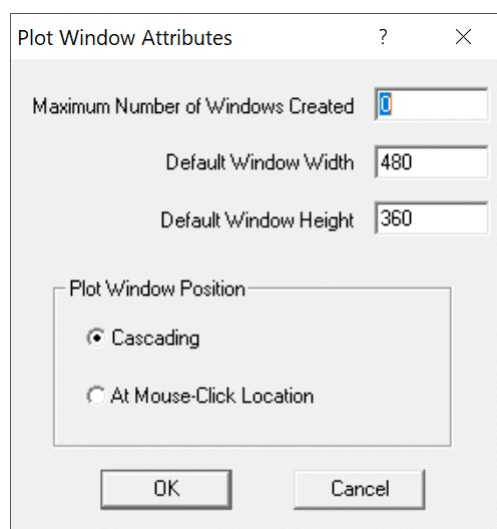
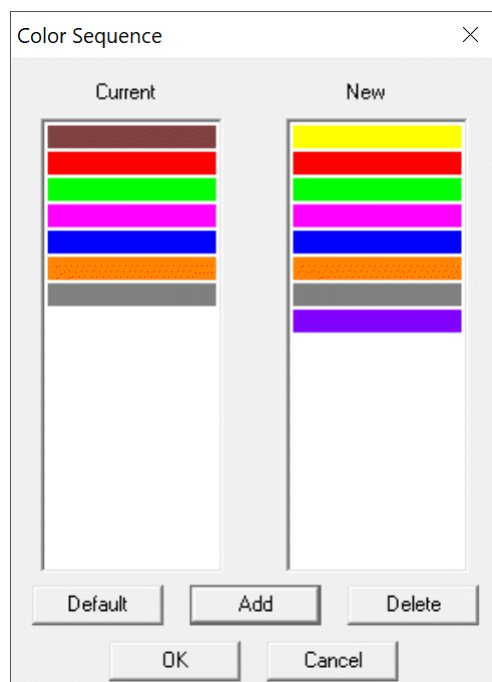


Chart - Color Sequence

Pirouette graphics adhere to a color scheme based on one of two criteria. If a class has been activated (see [“Activating a Class Variable”](#) on page 13-19), then coloring of sample points and traces corresponds to values in the activated class. If no class is activated, then coloring is based solely on the row or column index of the data plotted. In either case, the color mapping is drawn from the Color Sequence which can be modified via the Color Sequence preference item shown below.

Figure 16.51
Color Sequence
dialog box



Add or delete colors by clicking on the appropriate button in the dialog. Double-click on a color to change an existing color without affecting the order. Revert to Pirouette’s default colors by clicking on Default.

Prediction

The parameters in the dialog box shown in [Figure 16.52](#) allow some fine tuning during predictions. Details on how you should use these parameters are given in the respective algorithm chapters.

Figure 16.52
Prediction
Preferences dialog
box

The image shows a dialog box titled "Prediction Parameters" with two main sections: "Classification" and "Regression".

Classification Section:

- Probability: 0.950000 (0 - 1)
- Augment Sample Residual
- Mask Variable: Mask
- Calibration Transfer Type: None (dropdown menu)
- Window Size: 0
- Use class mean

Regression Section:

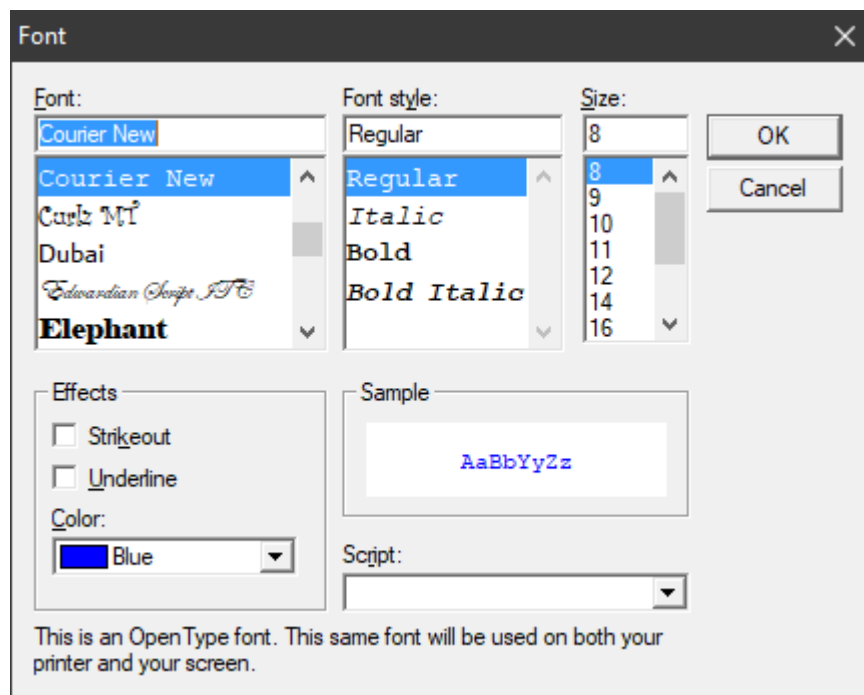
- Probability: 0.950000 (0 - 1)
- Mask Variable: Mask
- Calibration Transfer Type: None (dropdown menu)
- Window Size: 0

At the bottom of the dialog box are two buttons: "OK" and "Cancel".

Info Text

You can access an object's "historical" details via one of several info boxes. For example, model info is shown in the Predict Configure dialog box when you select a model's name. Attributes for this Info Text are set in a dialog box like that shown in [Figure 16.53](#).

Figure 16.53
Info Text dialog box



Because the information presented can be lengthy, specifying a small font insures that all text will fit in the space allotted.

Create/Delete/Load

It is sometimes advantageous to maintain several preference sets. The Create, Delete and Load items, which provide this management capability, are discussed in [“Preference Sets”](#) on page 10-21.

CASCADE/TILE

These items perform the function common to many Windows programs. Cascade arranges all open windows in a descending stack, each offset from the previous. Tile resizes and arranges all open windows such that they fill the available area but do not overlap.

Plot windows will be tiled in order of most recent interaction first, and the tiling order is down then across.

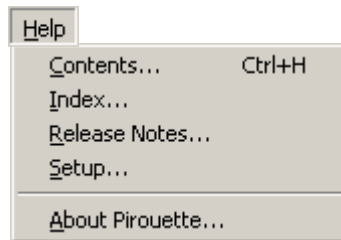
CLOSE WINDOW/CLOSE ALL WINDOWS

Close Window closes the active chart window; this is equivalent to clicking on the window’s close box. The chart can be redisplayed later by dragging and dropping from the Objects Manager. All existing chart windows can be closed at once by choosing Close All Windows. The Object Manager window cannot be closed, although it can be minimized.

Help Menu

The Help menu is your entry point to the on-line documentation supplied with Pirouette. The entire documentation set is included as portable document format (PDF) files which can be accessed from either the Help menu item or from the Information button in the ribbon.

Figure 16.54
Help menu



CONTENTS

The Contents item opens Acrobat Reader and loads the Pirouette User Guide. Using the built-in contents list (known as “bookmarks” in Reader), navigate from here to the specific chapter you want to examine.

INDEX

The Index item opens Acrobat Reader and loads the index to the Pirouette User Guide. Click on a page number associated with an index term to go that section of the main user guide document.

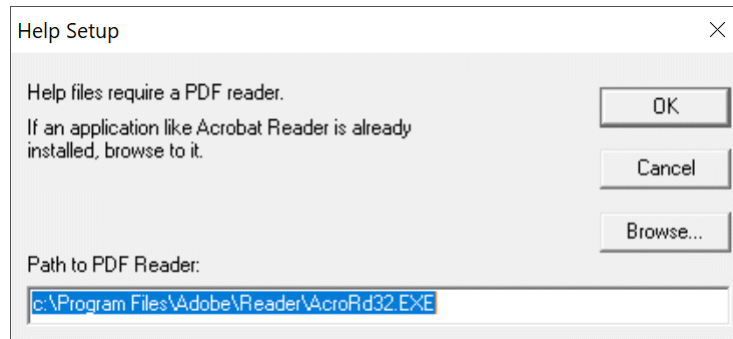
RELEASE NOTES

Pirouette is shipped with a separate file containing release notes. This contains information that did not make it into the main documentation as well as helpful information about the various versions and changes to Pirouette prior to this release.

SETUP

Pirouette comes with Acrobat Reader, in the event you do not already have a copy. Initially, Pirouette expects Reader to be in its default install directory. If you installed Reader into another directory, use Setup to locate it so that when you access the above help items, Pirouette will be able to start the Reader and show the help item. To modify the path to Reader, specify the path via the dialog box shown below.

Figure 16.55
Help Setup dialog
box

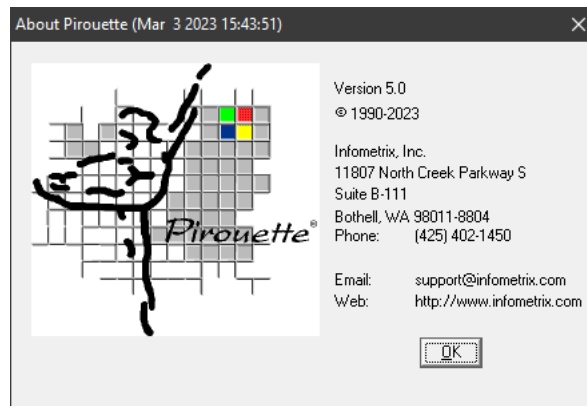


Note: *Pirouette's help was optimized for use with Acrobat Reader version 3.0 or later. If you use an earlier version of Reader, you may have difficulties printing to a non-postscript printer.*

ABOUT PIROUETTE

The About Pirouette dialog box contains the current version number as well as information on how to contact Infometrix. If needed, the build time is included in the title bar.

Figure 16.56
About Pirouette
dialog box



Part IV.

Appendices

17 An Introduction to Matrix Math

18 Tips and Troubleshooting

19 Pirouette Scripting

An Introduction to Matrix Math

Contents

Vectors and Matrices	17-1
Matrix Operations	17-3
Matrix Inversion	17-5
Eigenvectors and Eigenvalues	17-6
Reference	17-7

The fundamentals of matrix mathematics will be presented in this appendix for those readers for which this material is new or unfamiliar. These techniques, however, will not be discussed in great detail; if you find it necessary to probe more deeply, you should consult one of several treatises on the subject, one of which appears as a reference at the end of this appendix¹.

Notation used in this chapter, and in the Pirouette manual in general, will adhere to the following, typical set of rules. Scalars are represented by normal text; vectors are written as lower-case, bold characters; and matrices are shown as upper-case, bold characters. The transpose of a matrix, is denoted by an upper case, superscript ^T, while an inverse is shown by a superscript ⁻¹.

Vectors and Matrices

When several measurements are made on a sample, we can store those results as a vector, such as the row vector \mathbf{x} , written as:

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_m] \quad [17.1]$$

where the x_j would be the responses from the sample for m different variables.

Similarly, the results from making a single measurement for a collection of several samples could also be stored in a vector, in this case the column vector \mathbf{y} :

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \quad [17.2]$$

where the y_i are the responses for the n different samples for the measurement.

When many samples are characterized via the measurements of several variables, the responses can be represented in matrix form by building a composite of each of the sample's response vectors:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \quad [17.3]$$

In matrix notation, the dimensionality of the vector or matrix contains information about the number of samples and the number of measurements (*e.g.*, the dimensionality of the matrix in [equation 17.3](#) is $n \times m$). If the dimensions are the same, $n = m$, then the matrix is square.

There are a number of special (square) matrices whose use will be noted later. For example, a matrix all of whose values = 0, is termed a zero matrix:

$$\mathbf{X}_{\text{zero}} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad [17.4]$$

A matrix is called symmetric when the elements reflected through the diagonal are equal, *i.e.*, $x_{12} = x_{21}$, $x_{13} = x_{31}$. For example, the following matrix is symmetric:

$$\mathbf{A} = \begin{bmatrix} 4 & 1.3 & 22 \\ 1.3 & 9.5 & 0.68 \\ 22 & 0.68 & 0.14 \end{bmatrix} \quad [17.5]$$

A special case of a symmetric matrix is a diagonal matrix, in which all values for elements off of the diagonal are zero. Following is an example of a diagonal matrix:

$$\mathbf{B} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 9.5 & 0 & 0 \\ 0 & 0 & 0.14 & 0 \\ 0 & 0 & 0 & 15 \end{bmatrix} \quad [17.6]$$

The identity matrix \mathbf{I} is a special case of a diagonal matrix where all of the diagonal elements are ones. Thus:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad [17.7]$$

Finally, the transpose of a matrix is found by simply swapping the positions of the elements, e.g., x_{12} becomes x_{21} , x_{13} becomes x_{31} , and so on, as shown in the following equation:

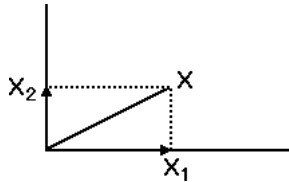
$$\mathbf{X}^T = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}^T = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \dots & \dots & \dots & \dots \\ x_{1m} & x_{2m} & \dots & x_{nm} \end{bmatrix} \quad [17.8]$$

Matrix Operations

In working with data expressed in vector and matrix notation, there are a number of linear algebra rules which govern the mathematics, many of which have analogies in scalar algebra.

A vector \mathbf{x} can be represented graphically as a line in m dimensions, whose endpoint is defined by a distance x_1 along the first axis, a distance x_2 along the second axis, and so on through the m axes. For example, the 2-dimensional vector \mathbf{x} (x_1, x_2) is shown in the following figure:

Figure 17.1
Graphical illustration
of a 2-dimensional
vector



An important characteristic of a vector, besides its orientation, is its length. We are familiar with the Pythagorean theorem which states that the length of the hypotenuse of a right triangle is the square root of the sum of the squares of the two sides of the triangle.

$$L_{\text{hypotenuse}} = (x_1^2 + x_2^2)^{1/2} \quad [17.9]$$

This formula can be generalized for any number of dimensions, such that the length L of any vector \mathbf{x} is computed as below. The notation $\|\mathbf{x}\|$ is often used for length.

$$L_{\mathbf{x}} = (x_1^2 + x_2^2 + \dots + x_m^2)^{1/2} \quad [17.10]$$

Multiplying a vector by a constant is done by multiplying each element in the vector by the constant:

$$c\mathbf{x} = [cx_1 \quad cx_2 \quad \dots \quad cx_m] \quad [17.11]$$

The length of the vector is increased accordingly:

$$L_{c\mathbf{x}} = c(x_1^2 + x_2^2 + \dots + x_m^2)^{1/2} = cL_{\mathbf{x}} \quad [17.12]$$

Scalar multiplication of a matrix is similar to that for vectors:

$$c\mathbf{X} = c \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} = \begin{bmatrix} cx_{11} & cx_{12} & \dots & cx_{1m} \\ cx_{21} & cx_{22} & \dots & cx_{2m} \\ \dots & \dots & \dots & \dots \\ cx_{n1} & cx_{n2} & \dots & cx_{nm} \end{bmatrix} \quad [17.13]$$

The sum of two vectors is found by summing the corresponding elements from each vector:

$$\mathbf{a} + \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \dots \\ a_n + b_n \end{bmatrix} \quad [17.14]$$

Sums of matrices are found in a similar fashion:

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{bmatrix} \quad [17.15]$$

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2m} + b_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \dots & a_{nm} + b_{nm} \end{bmatrix}$$

Matrix addition (and subtraction) is both commutative:

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A} \quad [17.16]$$

and associative:

$$(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C}) \quad [17.17]$$

Note, however, that for addition to be permitted, the dimensions of the matrices must be identical, *i.e.*, $n_a = n_b$ and $m_a = m_b$.

Two vectors are coincident (identical, other than length) if the angle between them is zero. We determine the angle from the following expression:

$$\cos(\theta) = \frac{\mathbf{x}^T \mathbf{y}}{L_{\mathbf{x}} L_{\mathbf{y}}} = \frac{\mathbf{x}^T \mathbf{y}}{(\mathbf{x}^T \mathbf{x})^{1/2} (\mathbf{y}^T \mathbf{y})^{1/2}} \quad [17.18]$$

where $\mathbf{x}^T \mathbf{y}$ is the inner product of \mathbf{x} and \mathbf{y} :

$$\mathbf{x}^T \mathbf{y} = x_1 y_1 + x_2 y_2 + \dots + x_m y_m \quad [17.19]$$

Note that the length of \mathbf{x} can also be expressed in the inner product notation, $\mathbf{x}^T \mathbf{x}$:

$$L_{\mathbf{x}} = (\mathbf{x}^T \mathbf{x})^{1/2} \quad [17.20]$$

From [equation 17.18](#) we see that if the inner product of \mathbf{x} and \mathbf{y} is zero then the cosine of the angle between \mathbf{x} and \mathbf{y} is also zero, therefore, the angle between the vectors must be 90° . Two vectors which form a right angle (90°) are perpendicular; in a data space of more than 2 dimensions, such vectors are said to be orthogonal.

A vector can be normalized to unit length ($\mathbf{x}^T \mathbf{x} = 1$) by dividing by its length:

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x}}{(\mathbf{x}^T \mathbf{x})^{1/2}} \quad [17.21]$$

Two normalized vectors which are also orthogonal are said to be orthonormal.

Matrix multiplication also requires compatibility in the dimensions of the matrices: the number of columns in the first matrix must equal the number of rows in the second. The dimensionality of the product matrix \mathbf{C} is determined by the number of rows in the first matrix \mathbf{A} and the number of columns in the second \mathbf{B} , respectively:

$$\mathbf{C}_{(n \times m)} = \mathbf{A}_{(n \times k)} \mathbf{B}_{(k \times m)} \quad [17.22]$$

Multiplication is carried out such that the element in the i th row and j th column in \mathbf{C} is formed from the inner product of the vectors from the i th row of \mathbf{A} and the j th column of \mathbf{B}

$$c_{ij} = \mathbf{a}_i \mathbf{b}_j = a_{i1} b_{1j} + a_{i2} b_{2j} + \dots + a_{ik} b_{kj} \quad [17.23]$$

Unlike matrix addition, matrix multiplication is normally not commutative (\mathbf{BA} would exist only if both \mathbf{A} and \mathbf{B} were square):

$$\mathbf{AB} \neq \mathbf{BA} \quad [17.24]$$

However, matrix multiplication is distributive and associative:

$$(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC} \quad [17.25]$$

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC}) \quad [17.26]$$

Multiplication of a matrix by the identity matrix of corresponding size leaves the matrix unchanged:

$$\mathbf{AI} = \mathbf{A} \quad [17.27]$$

Matrix Inversion

The inverse of a scalar is defined such that the product of the scalar and its inverse is 1:

$$k k^{-1} = 1 \quad [17.28]$$

The inverse of a matrix is defined similarly; the product of a square matrix and its inverse is the identity matrix:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} = \mathbf{A}^{-1}\mathbf{A} \quad [17.29]$$

For example, if matrix \mathbf{A} is the 2x2 matrix shown in the next equation:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad [17.30]$$

then its inverse is determined by:

$$\mathbf{A}^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \begin{bmatrix} a_{22}/\Delta & -a_{12}/\Delta \\ -a_{21}/\Delta & a_{11}/\Delta \end{bmatrix} \quad [17.31]$$

where Δ is the determinant of \mathbf{A} ,

$$\Delta = |\mathbf{A}| = a_{11}a_{22} - a_{12}a_{21} \quad [17.32]$$

Note that if the determinant is zero, then the inverse is not defined because a divide by zero would occur. Thus, for some matrices an inverse does not exist. Such matrices are termed singular matrices.

Diagonal matrices are invertible and are easily computed by taking the inverse of each element in the diagonal:

$$\mathbf{A}^{-1} = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}^{-1} = \begin{bmatrix} 1/a_{11} & 0 \\ 0 & 1/a_{22} \end{bmatrix} \quad [17.33]$$

If all of the columns (or rows) in a matrix are mutually orthogonal, then the inverse of the matrix is equivalent to its transpose:

$$\mathbf{N}^{-1} = \mathbf{N}^T \quad [17.34]$$

This concept provides an important tool for solving matrix equations.

Eigenvectors and Eigenvalues

For a square symmetric matrix \mathbf{A} , an eigenvector (and its associated eigenvalue) is defined as that vector \mathbf{x} for which the following relationship is true:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k &= \lambda x_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2k}x_k &= \lambda x_2 \\ &\dots \\ a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kk}x_k &= \lambda x_k \end{aligned} \quad [17.35]$$

or, in matrix form:

$$\mathbf{Ax} = \lambda \mathbf{x} \quad [17.36]$$

This can be rearranged to a form in which we can see how to solve for the roots of the equation:

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{x} = 0 \quad [17.37]$$

For each of the k dimensions of the matrix \mathbf{A} , there is an eigenvector and corresponding eigenvalue. Thus, \mathbf{A} is decomposed by the suite of eigenvectors:

$$\mathbf{A} = \lambda_1 \mathbf{x}_1 \mathbf{x}'_1 + \lambda_2 \mathbf{x}_2 \mathbf{x}'_2 + \dots + \lambda_k \mathbf{x}_k \mathbf{x}'_k \quad [17.38]$$

All of the k eigenvectors in the decomposition of \mathbf{A} are unique unless there are redundancies in the eigenvalues, *i.e.*, there are eigenvalues whose values are equal. If any of the eigenvalues are zero, their corresponding eigenvectors do not contribute to the sum, therefore, the term can be omitted from the expression with no loss of information.

Decomposition of a data matrix, such as just described, provides us with an alternate coordinate system with which to express our data. Instead of relating the individual data points to the original coordinates, the measured variables, we can transform the data points into a coordinate system in which the axes are the eigenvectors. Such a transformation normally results in an economy of expression: most of the dispersion in the data is compressed into the first few eigenvectors rather than being spread among all of the original measurements.

Reference

1. Malinowski, E.R. *Factor Analysis in Chemistry* - Second Edition, John Wiley & Sons, Inc.: New York (1991).

Tips and Troubleshooting

Contents

Tips	18-1
Frequently Asked Questions	18-2
Messages	18-3
Known Problems	18-12
Technical Assistance	18-14

We have tried to make Pirouette as reliable and intuitive as possible. Nevertheless, you may encounter situations in which an operation is unexpectedly terminated. This may occur due to a bug in the program or it may be the result of attempting to perform an operation using improper settings.

In the latter case, you will normally be presented with an error or warning message indicating the likely cause of the problem. These messages are included in this chapter with additional explanations for their cause.

When one of these errors occur, the dialog box which will be shown should give you a clue to its cause, as discussed below. Other situations may arise where you are unsure what to do next. We have tried to anticipate some of these questions; see [“Frequently Asked Questions”](#) on page 18-2.

Tips

The following ideas may help your productivity as you use Pirouette.

Saving Data

Starting with Pirouette 4.0 rev 1, the ability to run as a standard User has been enabled. However, such a User with limited permissions cannot write to folders in Program Files, the default path for Pirouette. Instead, save your files in My Documents or a sub-folder thereof.

Transposing data

Many data sources store their samples as vertical tables of numbers. There are two approaches to getting this data into row-oriented form for Pirouette.

1) The ASCII format which Pirouette can read allows some flexibility in the formatting structure. You can take advantage of this flexibility to read the transpose of the file. The

key is to place the #r (row) designator before the #c (column) designator. See “ASCII Files” on page 14-5 for more details. A step-by-step [procedure for preparing to transpose data](#) is given on our web site.

2) If you simply need to transpose one data file after loading and don’t need to be concerned about file formatting, use the Transpose menu item in the File menu (see “Transpose” on page 13-12).

Merging files from different directories

You can use the Windows Explorer Find or Search utility to make a list of all files of appropriate type. Be sure to restrict your search to the directory that contains only those subdirectories where your files reside. When the list of files is presented, highlight those files you wish to merge, then drag them onto the Pirouette interface. Switch to the Pirouette interface to respond to a dialog which asks whether to merge the files as samples or as variables.

Note that the order of loading of files grouped in this way will be in the order of presentation in the search folder (for example, by name or by date), with one caveat: the first file to load will be the one on which you click to initiate the drag into Pirouette.

Frequently Asked Questions

Most questions that you might have can probably be answered by a quick consultation of this document. The following discussions include answers to frequently asked questions (FAQs), which you should consult before seeking technical support. You should also check the “Known Problems” section of this chapter for a list of issues present in the current version. Finally, many other issues and applications are covered on the [Infometrix website \(https://infometrix.com/support/user-questions/\)](https://infometrix.com/support/user-questions/).

Why is my set name not enabled in the Predict configure dialog box?

The prediction algorithms in Pirouette are aware of excluded variables. Thus, if column exclusions were present in the training set, the algorithms will know to ignore those variables during prediction. This means that you cannot have column exclusions in the data for which you want to make a prediction. On the other hand, if there are no column exclusions, but only row exclusions, these subsets will appear enabled in the list.

I merged some data but the new rows (columns) are greyed out. How do I get them to be included?

Highlight the excluded rows and/or columns and do Edit > Include. Alternatively, drag the Disk icon from the Object Manager to the desktop. This creates a new subset that has all rows and columns included.

When saving a model, I type the model name and hit OK but nothing happens.

To save a model, you need to tell Pirouette which model(s) you intend to save. Since a Pirouette session can include an assortment of models, you must select a model by clicking on its name before the model save will occur.

I can’t see all of the text in the information box.

Depending on your screen resolution and the font which is set for the Info Box Text preference, certain information boxes (e.g., Model Info, Object Manager right mouse click

info) may not be large enough to show all of the data. With version 4.0, the Model Info text is placed within a scrollable window. Click on a text string, hold the mouse down, and drag downwards or to the right.

Workaround. Reduce the font size for the Info Box Text preference; Courier 9 points is usually small enough (see “Info Text” on page 16-43).

My screen is full of my results. How can I drag a result to make a new plot?

Because Pirouette allows you to view results from many analyses, it is not uncommon to fill the screen with chart windows. However, another valid target for a drag and drop operation is the ribbon itself. Note that the drop icon is the active cursor so long as you are over any blank Pirouette real estate, any window other than the Object Manager, or the entire ribbon area (see “Creating Charts from the Object Manager” on page 12-1).

My analysis aborted. What went wrong?

There are many reasons why an analysis might abort. To discover the most likely cause, click on the Details button (or double-click on the Abort message line) before you close the Run Status dialog box. The reason for the abort will be given in another message box. Examples of messages you may encounter are detailed later in this chapter.

When I try to save my data to a .PIR file, I get message “filename ... was not saved”.

If you drag a large number of files (more than ~45) from Windows Explorer into Pirouette, then try to save the file in Pirouette (*.PIR) format, this message will be presented. This is an artifact of the compiler used to control the Pirouette file server.

Workaround. Save the merged data into the ASCII (*.DAT) format instead. You can then load the ASCII file and save it to Pirouette format. You may have to quit Pirouette, restart and reload the ASCII file (tip: load it from the Recent Files list).

I selected Pirouette from the Start menu (or, I double-clicked on the Pirouette icon) but nothing happens. Pirouette does not run.

If you are running Windows and your account does not have at least Power User privileges, you will not be able to run Pirouette.

Additionally, your administrator will need to allow Write Permissions on your computer within the Pirouette folder [typically C:\Program Files\Infometrix\Pirouette #.##].

Messages

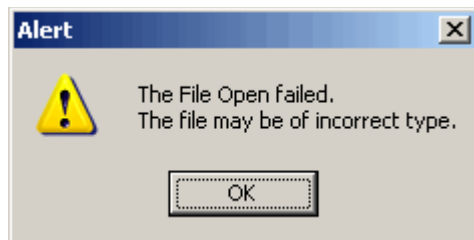
The messages which are displayed by Pirouette can be grouped into three types: Errors, Warnings and other forms of Alerts.

ERROR MESSAGES

Error messages are given when you and/or the program has done something wrong or unexpected. They are most likely to occur with certain operations, so the discussion which follows is grouped accordingly.

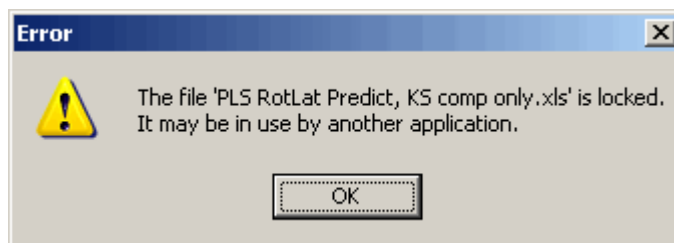
File Errors

Figure 18.1
File does not open



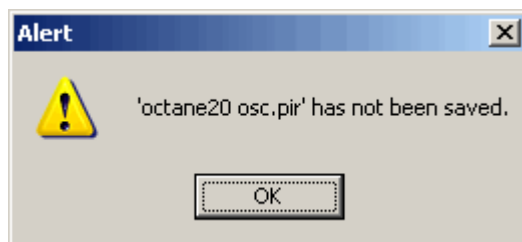
If Pirouette is not able to read the file you are trying to open, an alert will be displayed. Check to make sure that the proper file filter is specified. If you are trying to read an ASCII or a spreadsheet format file, open the file in the originating program or in a word processor to verify the format. Refer to “Opening and Merging Existing Data Files” in Chapter 14 for information on acceptable formats. If the file still will not read, check the Infometrix home page (<http://www.infometrix.com>) to see if there are updated file servers for the type you are trying, or contact the support line by phone or e-mail (support@infometrix.com).

Figure 18.2
File could not be opened



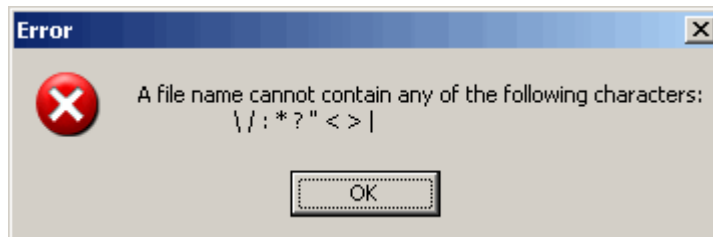
If a Pirouette file is already in use by another instance of Pirouette, you will not be allowed to open the file.

Figure 18.3
File could not be saved



Similarly, if you try to save a file that is in use by another application, the save will be blocked.

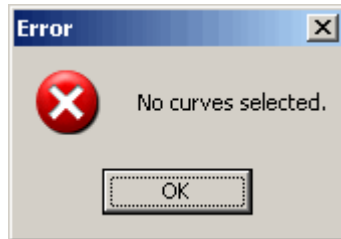
Figure 18.4
File names cannot contain illegal characters



Just as Windows Explorer will not allow certain characters to be included in a file name, so Pirouette will prevent you from attempting to create a file with illegal characters. If you encounter this error, remove the offending character and continue with the file save.

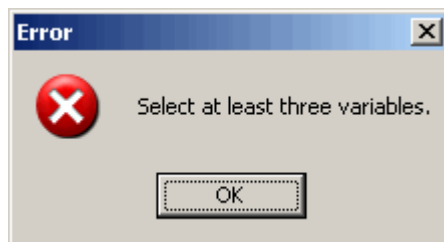
Plotting Errors

Figure 18.5
Line Plot is not shown



When viewing your data graphically, the Selector tool is used to add or remove data points from the view. A Pirouette graph must have at least one data point in a scatter presentation or one line in a line plot. If you try to remove all of the data from a plot, then hit the OK button, this message will be displayed.

Figure 18.6
Multiplot is not updated



If you use the Selector dialog box to modify the number of subplots to appear in a multiplot, you cannot select fewer than three items, otherwise an error message will be shown.

Figure 18.7
Scatter plot does not update

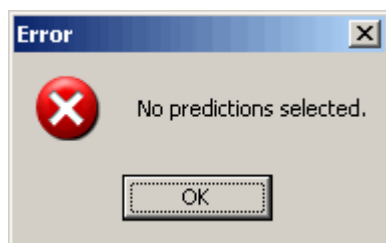


If you use the Selector dialog box to modify the axes to be shown in a scatter plot, you must specify the axis name correctly. Otherwise, an error message will be shown to let you know that your choice for the axis label is inappropriate.

Processing Errors

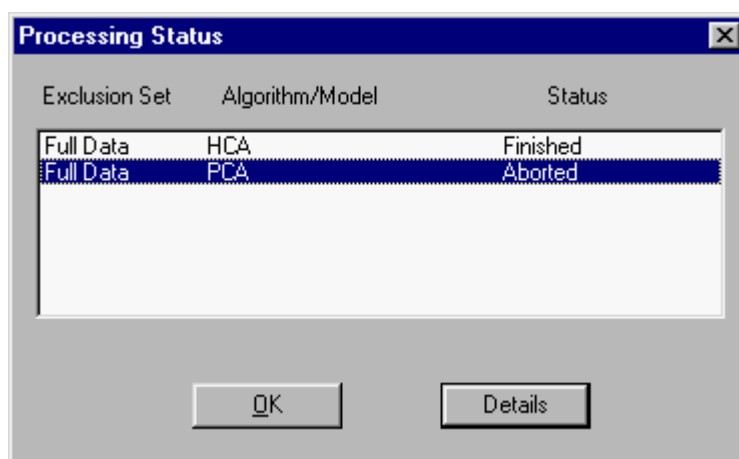
Unlike the Process > Configure dialog box, you must specify a subset and a model to perform a prediction. If you forget to select both, then hit the Run button, you will be warned that this combination was not properly selected. This would also occur if you pressed the Run button when no models were available or all subsets were disabled.

Figure 18.8
Subset - processing
pair not properly
selected



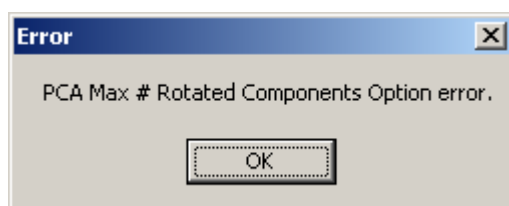
If an algorithm aborts during the run, the status box will not go away of its own accord and the run sequence will remain on screen for you to see which algorithms ran and which did not. To obtain the detail on why an algorithm aborts, double-click on the appropriate line or select the line and click on the Details button (see example figure below).

Figure 18.9
Processing Status
dialog showing an
aborted run



This action will bring up a message box, similar to the one in the following figure, describing the source of the problem.

Figure 18.10
Abort dialog box



Possible causes for an algorithm to abort are listed below.

Variable <name> is Invariant

One of the variables (variable number specified) is a constant. If variance or autoscaling are selected, but the values in a column are constant, a divide-by-zero could occur.

PCA Maximum Factors Option Error

The number of factors to be computed in PCA was set higher than is possible for the data set. Change the value to fall within the range given by the configuration parameters.

PCA has already been run

If an algorithm is configured to run with exactly the same parameters as in an analysis that has already been run, Pirouette will not recompute the results, rather it will redisplay the results for the equivalent analysis.

PCA Max # Rotated Components Option Error

You may not select to rotate more components in a Varimax operation than the number of factors set to compute in PCA.

Run Halted by User Abort

When a set of analyses are in progress, and the Status Box is still showing, you may click on the Abort button to prevent the completion of any analyses that are labeled as Waiting. The current analysis will complete.

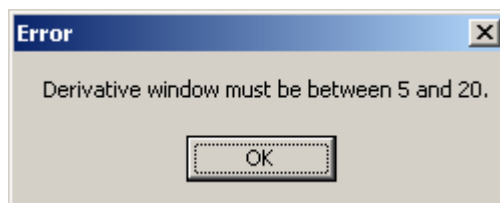
Data Matrix is Ill-Conditioned

If the true rank of your data matrix is less than the number of factors you requested to be extracted, this error can be generated. This situation can arise in a designed data set or in one in which there is very little variability among the samples or in a case when many variables are highly correlated or, in the extreme, they are identical except for a scale factor. Reduce the number of factors to be extracted and process again.

Transform Errors

If the settings for one or more of the transforms are not properly within the allowed range, an error will be generated and the analysis will abort. The error for the situation will be presented in a dialog box like that shown below. Possible transform errors are listed below

Figure 18.11
Example Transform
Error dialog

**Derivative window must be between 5 and 20**

You cannot set the number of points in a derivative to be larger than the number of variables in the full data set.

Smooth window must be between 5 and 20

You cannot set the number of points in a smooth to be larger than the number of variables in the full data set.

Cannot multiply by zero

You cannot set the multiplication factor to be zero, although the value can be negative or positive.

Must normalize by a positive real number

The normalization factor must be greater than zero.

Can only subtract on excluded variable

To use the Subtract transform, by variable, the variable which is selected must be excluded first. After subtraction, the variable selected will in effect become zero for all samples, permitting a divide-by-zero error. If excluded first, there will be no effect.

X missing value(s) in included rows

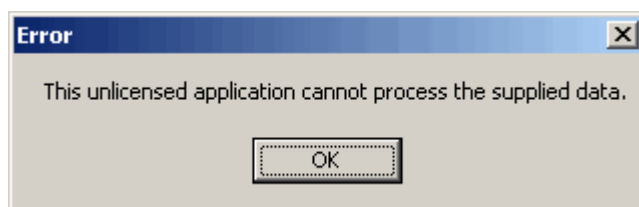
Some transforms operate exclusively on the included columns ignoring values in excluded variables. However, others, in particular the derivatives, use the entire variable range, even if some columns are excluded. Therefore, these transforms require that there be no missing values in the included rows.

WARNING MESSAGES

Warning messages are given to let you know that the operation you are about to make may create some undesirable effect. They will usually give you a chance to cancel the action to avoid the effect. As with the Error messages, these messages will occur mostly in certain scenarios, and are, therefore, listed in groups according to behavior.

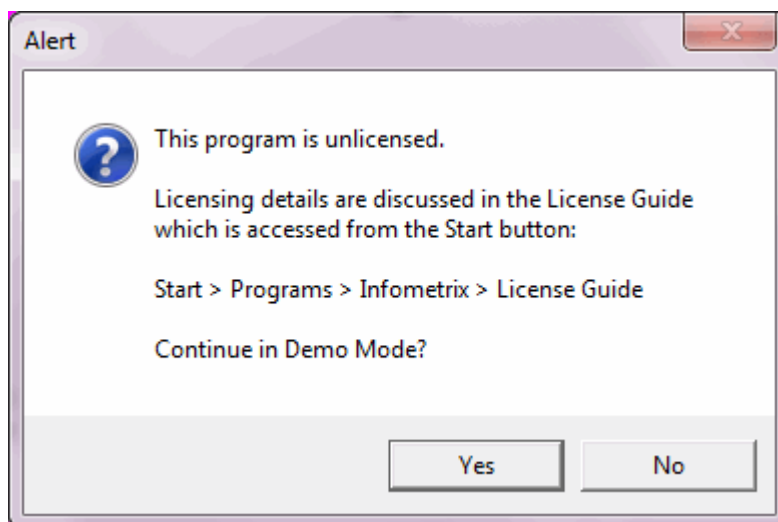
Demonstration Version Warnings

Figure 18.12
File does not open



If you try to process an unauthorized file in a demonstration version of Pirouette, this message will come to the screen. Pressing OK returns you to the Pirouette environment without opening the file.

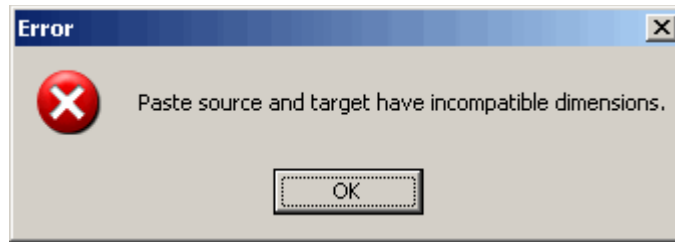
Figure 18.13
Demo version runs



If you try to run Pirouette without having first registered the software, this message will come to the screen. If you press the Yes button, the program will run, but in demonstration mode. Choosing No will exit Pirouette.

Delete and Edit Warnings

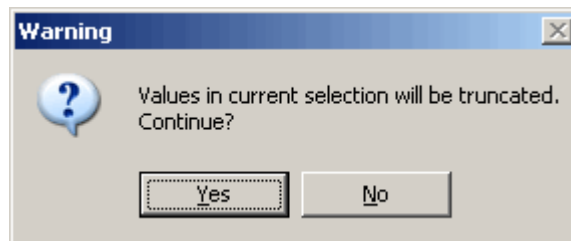
Figure 18.14
Paste does not complete



When pasting data into a Pirouette spreadsheet, the dimensionality of the target cell area, in terms of number rows and columns of data cells, must match that of the copied data. Two exceptions to this rule exist:

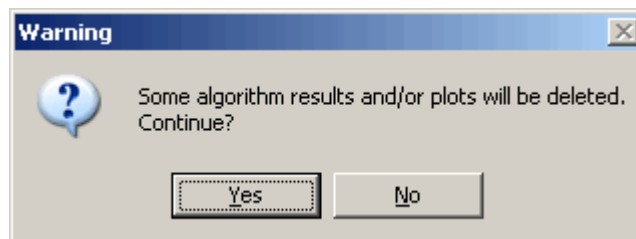
1. If only a single cell is selected
2. If the selected target cells represent an exact multiple of the copied cells

Figure 18.15
Converting a column to Class variable type



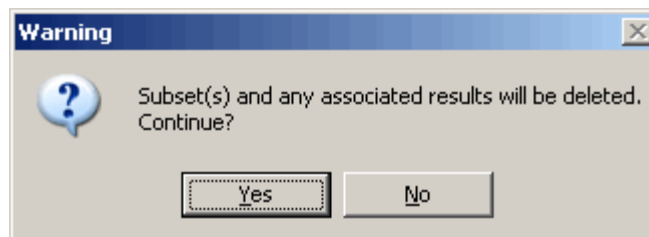
Because class variables are of nominal (integer) values only, when you convert a variable of another type to Class, the values will be truncated.

Figure 18.16
Editing data



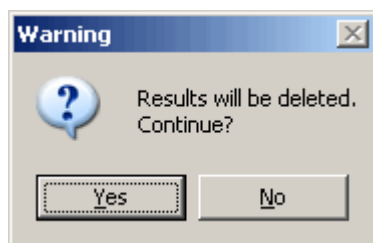
If you edit data—by typing, inserting or deleting—after results have been computed, the results affected will be lost.

Figure 18.17
Deleting a data subset



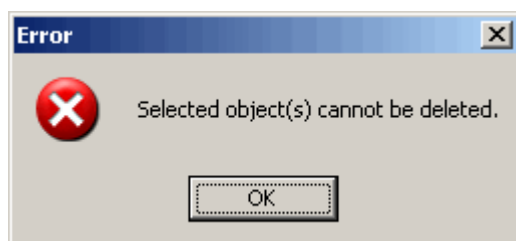
If you choose to delete a subset, from the Object Manager, not only will the subset be removed, but any associated results as well.

Figure 18.18
Deleting a result



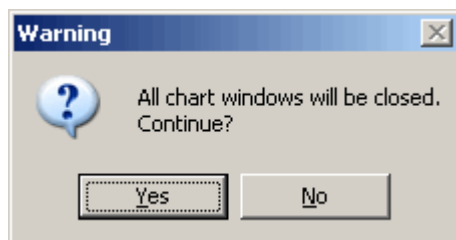
If you choose to delete a result, from the Object Manager, a warning will be given before proceeding so you can be sure of your action.

Figure 18.19
Cannot delete fixed
object



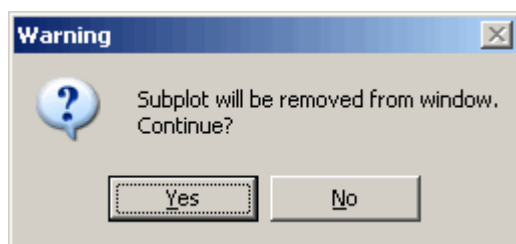
Certain objects in the Object Manager cannot be deleted, including individual computed results, the Disk icon which represents the data file, and the Chart icon itself. However, see below.

Figure 18.20
Deleting charts



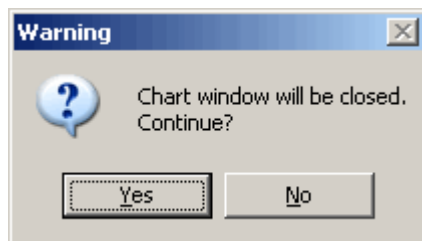
If you hit Delete when the Chart icon itself is selected, you will close all chart windows. This is a shortcut you should use carefully.

Figure 18.21
Delete a single
subplot



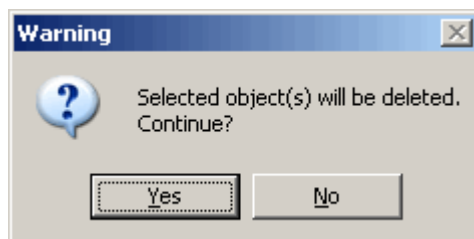
If, from the Object Manager Charts view, you hit Delete when a single subplot icon in a plot array is selected, only that plot will be removed.

Figure 18.22
Delete a single plot
or plot array



If you hit Delete when a single plot or plot array icon in the Object Manager is selected, only that plot or group of plots will be removed. In this way, you can remove an entire chart window or just selected subplots in a chart.

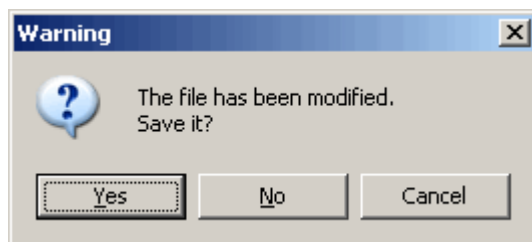
Figure 18.23
Delete several plots
or arrays



If you hit Delete when several plot or plot array icons (or a combination of the two) in the Object Manager are selected, that group of plots will be removed.

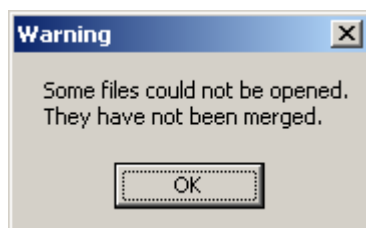
File Warnings

Figure 18.24
Saving a file



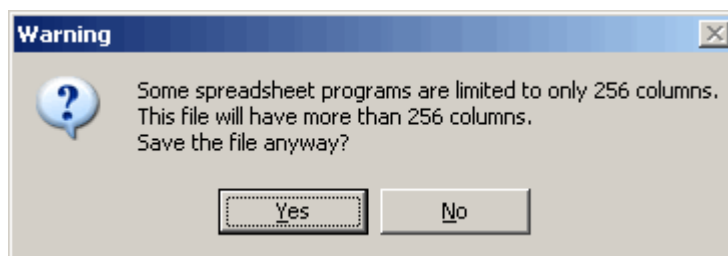
If you choose to read in a new data file or quit Pirouette, the program checks to see if you have saved your work. If you have performed any edits or computations since your last save, Pirouette will warn you of the potential loss and give you the option of saving the data or aborting the abandonment of calculated results.

Figure 18.25
Merging a file



Pirouette's Merge facility allows you to merge one or more files, as either new samples or variables. If one or more of the files to be merged cannot be read by Pirouette, a warning will be issued.

Figure 18.26
Saving a file with
more than 256
columns

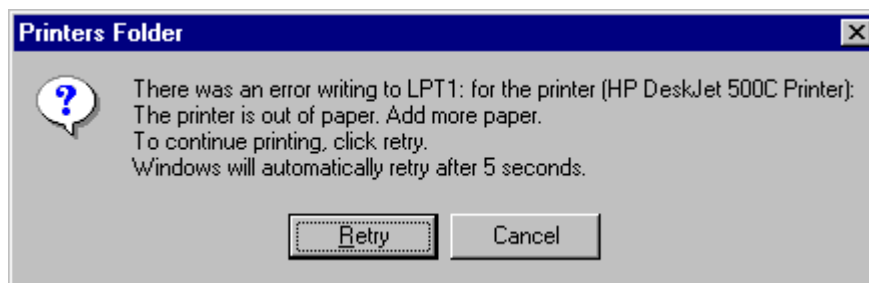


If you request to save a file with more than 256 variables to an old Excel spreadsheet format (.XLS), you can still load that file back into Pirouette, but the extra columns may become lost if you open the file in an older spreadsheet application.

OTHER ALERTS

Besides the Error and Warning messages, other alerts may be given; some of these are generated by the Windows operating system.

Figure 18.27
Printing does not
happen



If your printer is not attached or off-line when you ask Pirouette to print, you will see a system message as in the figure. Verify that your printer is on and attached to your computer, then try the print task again.

Known Problems

The following are known issues or problems in this version of Pirouette. Contact Info-metrix for the status of the fixes for these problems.

- When loading a data file with blank sample and/or variable names (e.g., from an Excel file), Pirouette will apply default names. However, a scatter plot may not initially show names for all points. Scroll through the entire table so that the names get created, then show the scatter plot.
- If you change to a preference set which contains different color assignments in the Color Sequence, some plots may not properly update their colors. Either open the Color Sequence dialog box and click OK or close the plot and reopen a new copy.
- Some users have reported issues following multiple insertions and/or deletions. As a workaround, save the file in the ASCII (*.dat) format, load this new file, then reprocess.
- Repeatedly magnifying line plots can occasionally cause spurious horizontal or vertical lines to appear which vanish when the plot is unmagnified. This problem derives from your system's graphics card.

- When using SIMCA with Scope set to Global, some objects may be computed improperly. We recommend that you use only Local Scope, the option which is most reasonable in the majority of situations.
- Merging files with more than 65000 variables may fail unexpectedly if more than a few samples are selected. If possible, merge one sample at a time and save intermediate files.

Technical Assistance

Infometrix offers readily available technical assistance. We can provide you with assistance if you are having difficulty in installing or running the software. If you require assistance on the use of the technology to solve particular problems, Infometrix can also provide consulting and/or training in general or customized to your application. Contact us for more details.

Telephone: (425) 402-1450

Email: support@infometrix.com

Current applications information, as well as a conduit to a variety of other sites for chemometric information, is available in the Infometrix web page:

<https://www.infometrix.com/>

Pirouette Scripting

Contents

Introduction	19-1
Rules and Instructions	19-2
Scripting Commands - Alphabetical Order	19-3
Scripting Commands - Functional Order	19-6
Example Scripts	19-8

Introduction

The Pirouette user interface is easy to interact with, containing many shortcuts for users who like to use them. There are times when you may want to perform some well-defined routine processing to your data, with no interaction, in an effort to be efficient. Pirouette 5 has been enhanced to allow scripting of common functions.

Scripting for Pirouette does not use any language as a basis, rather, it consists of a set of predefined command codes. Script files are composed as a set of these codes. There are rules for the presentation of the script codes and there is an implicit order of operation in the sequence of commands.

The script files can be introduced to Pirouette in one of two ways:

- As a command line parameter
- As a key in the Infometrix.ini file

When invoked from the command line, the script file must use .pscript as the file extension. Here is an example of calling a script from the Run dialog:

```
C:\PROGRAM FILES (X86)\INFOMETRIX\PIROUETTE 5.0\PIROU-
ETTE.EXE "C:\PROGRAMDATA\INFOMETRIX\SCRIPTS\MY-
SCRIPT.PSCRIPT
```

Next is an example of invoking a script using the ini file, found here: C:\Windows\infometrix.ini. The script file name is set as a value for the Run Script key under the [Pirouette] block in the ini file.

```
[Infometrix Locations]
Common Files=C:\Program Files (x86)\Common Files\Infometrix\
License2=C:\Program Files (x86)\Common Files\Infometrix\Licensing
```

```
LogFile=C:\Program Files (x86)\Common Files\Infometrix\Logging
Pirouette 5.0=C:\Program Files (x86)\Infometrix\Pirouette 5.0\
[Pirouette]
Run Script = "C:\ProgramData\Infometrix\Scripts\myscript.pscript"
```

There is no error checking when running Pirouette with a script. If it fails, you will not be notified with a reason for the failure. In some instances, if the presentation of the script results in an actual bug, there may be a reason given in the log. Otherwise, caveat emptor.

Rules and Instructions

Each line of a script file must contain a single character known as the Command, a mandatory single space, and a Parameter (the interpretation of which varies with the Command). ALL characters on a line after the first space are defined as the Parameter Line.

Every line of the script is meaningful. Blank lines are not supported.

There is no white space in any Parameter Line. Everything counts. Sometimes multiple values ('Parameters') are combined within a Parameter Line. When this is done, a single space (and only a space) separates each successive individual parameter. There should not be any spaces at the end of the Parameter Line.

Commands ARE case sensitive. a and A have different meanings.

Parameters are required, even if they have no meaning. There MUST be at least one visible character in the Parameter Line of each and every line of the script.

Do not use quotation marks of either type for any reason.

Text Bitmaps (whether in the script or in a separate file) are a string of nothing but 0s and 1s. No white spaces, no line breaks. The number of characters (0/1) must match exactly with a data file's number of Columns or Rows.

The commands in a script file are executed in order, before the Pirouette user interface becomes active. Once the user interface is active, there are no further script commands and no provision to invoke script commands.

Order of Commands matters. You need to load a file, for example, before running an algorithm.

There can be ONLY ONE algorithm run declared within a script. It uses the last exclusion set created, the cumulative algorithm parameters set, and the cumulative transform options declared (in the order declared). The only way to reset the transforms is to quit. The algorithm options may be set any number of times, but only the last setting is used. If you need to run another algorithm on a file, prepare a separate script that you would invoke after quitting Pirouette (either via the script or manually).

Success or failure of script commands is (hopefully) indicated in the Infometrix Log. There is no user interface while scripting. Scripting cannot be prepared from the Pirouette interface, only from a text writing program.

Scripting Commands - Alphabetical Order

Cmd	Parameter	Description and Options
1	integer	1st Derivative Transform. Parameter indicates # of points.
2	integer	2nd Derivative Transform. Parameter indicates # of points.
a	integer integer	Align Transform. 1st Parameter is Window size. 2nd Parameter is Align to row #. There must be (exactly) 1 space between parameters.
A	char	Declare regression Algorithm to use. Parameter indicates type: S Partial Least Squares Regression C Principal Components Regression
b	char integer	Baseline Correct Transform. 1st parameter indicates type, with: S Subtract Sample. 2nd parameter indicates Row #. L Linear Fit. 2nd parameter indicates Row # as mask IF non-0. Q Quadratic Fit. 2nd parameter indicates Row # as mask IF non-0. C Cubic Fit. 2nd parameter indicates Row # as mask IF non-0. For the later three cases, a 0 integer indicates no Row number. There must be (exactly) one space between parameters. Both are required.
B	integer	Set Best1 (do before alg Run 'G' command). Parameter indicates Best1.1 Note that not all algorithms or all situations take a scalar Best1. Some may require a vector of values. Only integer values are presently supported.
c	integer	MSC Transform. Parameter indicates Row Index [1..n]. Use '0' for none.
C	text bitmap	Declare Column Selection via in-script string of 0 and 1. (1 indicates an exclusion.)
d	char integer	Divide By Transform. 1st parameter indicates type, with: 1 Sample 1-Norm. 2nd parameter indicates Row # as Mask IF non-0. 2 Sample 2-Norm. 2nd parameter indicates Row # as Mask IF non-0. M Sample Max. 2nd parameter indicates Row # as Mask IF non-0. R Sample Range. 2nd parameter indicates Row # as Mask IF non-0. V Value at Variable. 2nd parameter indicates variable #. S Sample Vector. 2nd parameter indicates Row # to use. For the first four cases, a 0 integer indicates no Row # as mask. There must be (exactly) one space between parameters. Both are required.
D	char	Distance Metric. Parameter is one of: E Euclidean N Euclidean (no init)
E	exclusion set	Create an exclusion set using last declared column & row. Set is named in the Parameter.
F	file path+	Save a Pir2 file. File is named in the Parameter. Existing files of the same name will not be overwritten.
G	alg name	Run an algorithm. Algorithm is named in the Parameter and is an EXACT string match with Pirouette internals. Presently supported: ALS, CLS, HCA, KNN, LWR, MCR, PCA, PCR, PLS, PLS-DA, SIMCA
I	integer	Index (1-offset) of class variable to use. (1st class variable is '1', 2nd '2', etc)
I		Log10 Transform. Note that a parameter is required (but meaningless).
K	char	Linkage Method. Parameter is one of: S Single D centroid C Complete I Incremental

19 Pirouette Scripting: Scripting Commands - Alphabetical Order

Cmd	Parameter	Description and Options
		M Median
		A group Average
		F Flexible
L	file path+	Load a data file.
M	file path+	Save a model file.
n	float	Normalize Transform. Parameter is normalize value.
N	integer	Number of Neighbors.
O	char	Set Closure. Parameter is one of: N none A Amounts P Profiles
P	float	Probability Threshold. Parameter should be < 1.00.
Q	integer	Set the 'Compute Q-threshold' flag. Integer is interpreted as a boolean value: 0 for FALSE non-0 for TRUE
R	text bitmap	Declare Row Selection via in-script string of 0 and 1. (1 indicates an exclusion.)
s	integer	Smooth Transform. Parameter is number of points.
S	file path+	Declare Row Selection via file. Files are text 'bitmaps' of 0 and 1. (1 indicates exclusion.)
T	char	Dendrogram Orientation. Parameter is one of: S Sample V Variable
U	2chars <2chars>	Unimodality. Parameters may appear in any order. They require 1 space (exactly) between them if both are included. 1st & 2nd parameter one of: A0 Amounts 'false' A1 Amounts 'true' P0 Profiles 'false' P1 Profiles 'true'
v		SNV Transform. Parameter is required, though ignored.
V	file path+	Declare Column Selection via file. Files are text 'bitmaps' of 0 and 1. (1 indicates an exclusion.)
W	char	Scope. Parameter is one of: G Global L Local
X		Exit Pirouette. Parameter is required, though ignored.
~	char	Set the Initial Estimate from (parameter): R rows C columns
!	char	Declare Algorithm Preprocessing. Characters recognized are: A(utoscale) M(ean center) P(areto) R(ange scale) V(ariance scale)
#	integer	Declare MaxFactors for Algorithm.
&	char<integer>	Declare ValidationMethod for Algorithm. Characters recognized are: C lass Variable) [Number indicates a class column index] S tep validation) [Number indicates Leave Out number]

Cmd	Parameter	Description and Options
		X (cross validation) [Number indicates Leave Out number] Use '0' for none or N/A.
*	char<integer>	Declare RotationMethod for Algorithm. Characters recognized are: R(aw) [Number indicates Max Rotated Factors] N(ormal) [Number indicates Max Rotated Factors] W(eighted) [DO NOT include a number or any other input] W(eightedNumbers) [Number indicates Max Rotated Factors] Use '0' for No Rotation
-	char float/int	Subtract Transform. 1st parameter indicates type with: # (value) [float 2nd parameter indicates value to subtract] V(ariable) [int 2nd parameter indicates variable number to subtract] There must be (exactly) one space between parameters. Both are required.
+	2chars <2chars>	Non-Negativity. Parameters may appear in any order. They require 1 space(exactly) between them if both are included. 1st & 2nd parameter one of: A0 Amounts 'false' A1 Amounts 'true' P0 Profiles 'false' P1 Profiles 'true'

Key



indicates an optional value.

1

algorithm does not presently support setting 'Best I'.

Scripting Commands - Functional Order

Cmd	Parameter	Description and Options
G	alg name	Run an algorithm. Algorithm is named in the Parameter and is an EXACT string match with Pirouette internals. Presently supported: ALS, CLS, HCA, KNN, LWR, MCR, PCA, PCR, PLS, PLS-DA, SIMCA
X		Exit Pirouette. Parameter is required, though ignored.
F	file path+	Save a Pir2 file. File is named in the Parameter. Existing files of the same name will not be overwritten.
L	file path+	Load a data file.
M	file path+	Save a model file.
C	text bitmap	Declare Column Selection via in-script string of 0 and 1. (1 indicates an exclusion.)
V	file path+	Declare Column Selection via file. Files are text 'bitmaps' of 0 and 1. (1 indicates an exclusion.)
R	text bitmap	Declare Row Selection via in-script string of 0 and 1. (1 indicates an exclusion.)
S	file path+	Declare Row Selection via file. Files are text 'bitmaps' of 0 and 1. (1 indicates an exclusion.)
E	exclusion set	Create an exclusion set using last declared column & row. Set is named in the Parameter.
!	char	Declare Algorithm Preprocessing. Characters recognized are: A(utoscale) M(ean center) P(areto) R(ange scale) V(ariance scale)
*	char<integer>	Declare RotationMethod for Algorithm. Characters recognized are: R(aw) [Number indicates Max Rotated Factors] N(ormal) [Number indicates Max Rotated Factors] W(eighted) [DO NOT include a number or any other input] W(eightedNumbers) [Number indicates Max Rotated Factors] Use '0' for No Rotation
Q	integer	Set the 'Compute Q-threshold' flag. Integer is interpreted as a boolean value: 0 for FALSE non-0 for TRUE
A	char	Declare regression Algorithm to use. Parameter indicates type: S Partial Least Squares Regression C Principal Components Regression
B	integer	Set BestI (do before alg Run 'G' command). Parameter indicates BestI. 1 Note that not all algorithms or all situations take a scalar BestI. Some may require a vector of values. Only integer values are presently supported.
#	integer	Declare MaxFactors for Algorithm.
&	char<integer>	Declare ValidationMethod for Algorithm. Characters recognized are: C(lass Variable) [Number indicates a class column index] S(tep validation) [Number indicates Leave Out number] X (cross validation) [Number indicates Leave Out number] Use '0' for none or N/A.
I	integer	Index (1-offset) of class variable to use. (1st class variable is '1', 2nd '2', etc)
N	integer	Number of Neighbors.
P	float	Probability Threshold. Parameter should be < 1.00.

Cmd	Parameter	Description and Options
W	char	Scope. Parameter is one of: G Global L Local
O	char	Set Closure. Parameter is one of: N none A Amounts P Profiles
U	2chars <2chars>	Unimodality. Parameters may appear in any order. They require 1 space (exactly) between them if both are included. 1st & 2nd parameter one of: A0 Amounts 'false' A1 Amounts 'true' P0 Profiles 'false' P1 Profiles 'true'
~	char	Set the Initial Estimate from (parameter): R rows C columns
+	2chars <2chars>	Non-Negativity. Parameters may appear in any order. They require 1 space(exactly) between them if both are included. 1st & 2nd parameter one of: A0 Amounts 'false' A1 Amounts 'true' P0 Profiles 'false' P1 Profiles 'true'
D	char	Distance Metric. Parameter is one of: E Euclidean N Euclidean (no init)
K	char	Linkage Method. Parameter is one of: S Single D centroid C Complete I Incremental M Median A group Average F Flexible
T	char	Dendrogram Orientation. Parameter is one of: S Sample V Variable
1	integer	1st Derivative Transform. Parameter indicates # of points.
2	integer	2nd Derivative Transform. Parameter indicates # of points.
s	integer	Smooth Transform. Parameter is number of points.
b	char integer	Baseline Correct Transform. 1st parameter indicates type, with: S Subtract Sample. 2nd parameter indicates Row #. L Linear Fit. 2nd parameter indicates Row # as mask IF non-0. Q Quadratic Fit. 2nd parameter indicates Row # as mask IF non-0. C Cubic Fit. 2nd parameter indicates Row # as mask IF non-0. For the later three cases, a 0 integer indicates no Row number. There must be (exactly) one space between parameters. Both are required.
d	char integer	Divide By Transform. 1st parameter indicates type, with: 1 Sample 1-Norm. 2nd parameter indicates Row # as Mask IF non-0.

19 Pirouette Scripting: Example Scripts

Cmd	Parameter	Description and Options
		2 Sample 2-Norm. 2nd parameter indicates Row # as Mask IF non-0. M Sample Max. 2nd parameter indicates Row # as Mask IF non-0. R Sample Range. 2nd parameter indicates Row # as Mask IF non-0. V Value at Variable. 2nd parameter indicates variable #. S Sample Vector. 2nd parameter indicates Row # to use. For the first four cases, a 0 integer indicates no Row # as mask. There must be (exactly) one space between parameters. Both are required.
n	float	Normalize Transform. Parameter is normalize value.
c	integer	MSC Transform. Parameter indicates Row Index [1..n]. Use '0' for none.
v		SNV Transform. Parameter is required, though ignored.
-	char float/int	Subtract Transform. 1st parameter indicates type with: # (value) [float 2nd parameter indicates value to subtract] V(ariable) [int 2nd parameter indicates variable number to subtract] There must be (exactly) one space between parameters. Both are required.
l		Log10 Transform. Note that a parameter is required (but meaningless).
a	integer integer	Align Transform. 1st Parameter is Window size. 2nd Parameter is Align to row #. There must be (exactly) 1 space between parameters.

Key

<>

indicates an optional value.

1

algorithm does not presently support setting 'Best I'.

Example Scripts

LoadRunPCA.pscript

Load a file, set up parameters, run PCA.

Script commands	Explanation
L C:\Program Files (x86)\Infometrix\Pirouette 5.0\Data\XCIP4.DAT	Load the file XCIP4.DAT
! M	Set preprocessing to mean-center
# 10	Set maximum factors to 10
B 0	Don't set optimal factors; let Pirouette do so
G PCA	Run PCA

LoadRunPLSSaveQuit.pscript

Load a file, set up parameters, run PLS, save a Pirouette file, exit.

Script commands	Explanation
L C:\Program Files (x86)\Infometrix\Pirouette 5.0\Data\HYDROCRB.DAT	Load the file HYDROCRB.DAT
! M	Set preprocessing to mean-center
# 10	Set maximum factors to 10
B 0	Don't set optimal factors
G PLS	Run PLS

Script commands	Explanation
F C:\Program Files (x86)\Infometrix\Pirouette 5.0\Data\HYDROCRBsave.PIR2	Save file to a Pirouette 2 format
X 0	Quit Pirouette

LoadExcludeSNVPLS.pscript

Load a file, set up an exclusion bitmap for samples, give a name to the exclusion set, set transform to use SNV, set up parameters, run PLS.

Script commands	Explanation
L C:\Program Files (x86)\Infometrix\Pirouette 5.0\Data\HYDROCRB.DAT	Load the file HYDROCRB.DAT
R 00001000000000000000000000000000	Define a row bitmap; 1 means exclude this row
E Exclude5	Set exclusion set name to "Exclude5"
v 0	Set transform to SNV
! M	Set preprocessing to mean-center
# 10	Set maximum factors to 10
B 0	Don't set optimal factors
G PLS	Run PLS

LoadExcludeRCxValSNVPLS.pscript

Load a file, set up exclusion bitmaps for rows and for columns, name the exclusion set set up parameters including leave 4 out cross validation, run PLS.

Script commands	Explanation
L C:\Program Files (x86)\Infometrix\Pirouette 5.0\Data\HYDROCRB.DAT	Load the file HYDROCRB.DAT
R 00001000000000000000000000000000	Define a row bitmap; 1 means exclude this row
V C:\ProgramData\Infometrix\Scripts\HydrocrbExcludeVars.txt	Define a column bitmap from a file;
excludes some x and y variables	
E ExcludeRowAndColumns	Set exclusion set name to "ExcludeRowAndColumns"
v 0	Set transform to SNV
! M	Set preprocessing to mean-center
& X4	Set cross validation, with 4 left out
# 20	Set maximum factors to 20
B 5	Set optimal factors to 5
G PLS	Run PLS

Index

Numerics

- 2D scatter plot
 - Axes 12-5
 - Interaction 12-5
- 3D scatter plot
 - Axes 12-5
 - Preferences 10-12
 - Spinning
 - Arrow keys 12-10
 - Cylindrical vs. spherical 12-11
 - Depth cueing 12-11
 - Momentum spinning 12-10
 - Spin control buttons 12-10
 - Spinner Tool 12-9

A

- Abort
 - Details 16-31
 - Message 18-6
- Abstract factor 5-14
- Acrobat Reader
 - using for on-line Help 16-45
- Activate class 12-27, 13-19, 16-15
- Activate Class button 12-35
- Active class 6-2, 6-27, 7-44
- AIA file format 4-23, 14-10, 15-5
- Alert messages 18-12
- Align
 - Discussion 4-22
 - for Chromatography 4-22
 - Options 4-23
- Alt key 16-1
- Alternating Least Squares (ALS) 8-3
- Appending data 16-5
- ARCH 5-10, 9-2
- Arrow
 - Cursor 10-6
 - in Scroll Bar 13-4
- ASCII
 - File input 14-5
 - File output 15-3
 - Models 15-9
- Asterisk (*)
 - Class variable flag 14-8
 - Filling missing values 13-13
 - Missing value flag 16-6
- Autoscale 4-29

Axis

- Default label 10-17
- Selection 10-5, 12-5

B

- Baseline Correction 4-18
- Bidiagonalization 7-5
- Bitmap 15-2, 16-13
- Block scaling 4-17
- Bounding ellipse 6-24
- Buttons
 - Ribbon 10-3

C

- Calibration transfer 4-33, 6-29, 7-56
- Category validation 5-21
- Centroid link 5-3, 5-8
- Centroidal clustering 5-6
- Charts
 - Creating 12-1, 12-3
 - Custom 11-8
 - Getting information 12-2
 - Graph types 12-4
 - Label preferences 10-16
 - Removing 11-7
 - Window preferences 10-16
 - Window titles 12-3
- Chemometrics
 - Information 1-6
- Chromatographic alignment 4-22
- Class
 - Active class cue 3-15
 - Color mapping to 12-34
 - Distances 6-22
 - Fit 6-9
 - Probabilities (SIMCA) 6-27
 - Projections 6-24
 - Variable 13-5
- Classical Least Squares (CLS)
 - Math 7-44
 - Options 7-48
 - Prediction 7-53
- Clear 13-9
- Click-drag 10-1, 13-4
- Cloaking 12-8
- Closure 4-16
- Clustering

- Centroidal 5-6
- Farthest neighbor 5-5
- Nearest neighbor 5-5
- Color
 - and Dendrogram 12-23
 - Mapping from class 12-34
 - of Lines 12-15
 - of Text 10-8
 - Preferences 10-7
 - Sequence 10-18
- Column
 - Index 13-19
 - Types 13-5
- Communality 5-28
- Complete link 5-3, 5-7
- Confidence ellipse 5-35
- Confidence limits 6-24, 7-12, 7-35
- Confusion matrix 7-39
- Contributions 5-26, 5-40, 5-46
- Control-click 10-2
- Copy
 - Data values 13-9
 - Graphics 15-2
 - Results 16-13
- Correlation spectrum 7-26
- Cross validation 5-20, 7-46
- Crossover 5-9
- Ctrl-click 16-5
- Current cell 13-15
- Cursor descriptions 10-6
- Curve resolution 8-1, 8-12
- Cut, Copy, Paste 16-13

- D**
- DAT file 1-2, 14-5
- Data point
 - Size 10-11
- Data scaling 12-11
- Data sets
 - ALCOHOL 9-2
 - ARCH 5-10, 9-2
 - DAIRY 9-2
 - DIESEL 9-3
 - FUEL 9-3
 - HYDROCRB 9-3
 - MNAPS 9-4
 - MYCALIGN 9-4
 - MYCOSING 9-4
 - OCT_TEST 3-29
 - OCTANE20 3-4, 5-11, 9-4
 - PALMCHRO 14-3
 - RANDOM 5-4, 9-5
 - SEVEN 9-5
 - TERNARY 9-5
 - XRF 9-5
- Decision diagram 6-10, 6-23
- Defaults
 - Preferences 10-20
 - Subset name 11-6
- Delete
 - Charts 11-7
 - Data values 13-9
 - Objects 11-7
 - Samples, variables 16-14
- Demonstration version 18-8
- Dendrogram
 - Activating a class 12-27
 - Arrow keys in 12-25
 - Description 5-1, 12-22
 - Navigation 12-25
 - Similarity 12-23, 12-25
- Dependent variable 4-5, 7-1, 7-3, 13-5
- Derivative 4-12
- Determinant (of a matrix) 17-6
- Diagonal matrix 17-2
- Diamond marker 7-16
- DIESEL 9-3
- Dimensionality 5-14, 14-6
- Direct Standardization 4-35
- Discriminant Analysis 6-1
- Discrimination Power 6-21
- Disk icon 11-9
- Display Menu 16-34
- Displaying results 10-17
- Distance
 - Between classes (SIMCA) 6-21
 - Euclidean 6-3
 - Leverage 7-11
 - Metric 5-2
 - Prediction (SIMCA) 6-22
 - Similarity 5-2
- Divide by
 - Options 4-14
 - Subset mean 4-17
 - Vector range 4-15
- Dollar sign (\$) 14-8
- Drag and drop
 - Charts 12-2, 12-3
 - Data files 14-5
- Drop button 12-3

- E**
- Edit
 - Menu 16-11
 - Tools 10-4
- Eigenvalues 5-18, 5-22, 7-15, 17-7
- Eigenvector 5-14, 17-6
- Elevator box 13-4
- Email address 18-14
- Enhanced Meta File (EMF) 15-2, 16-14
- Error Analysis 7-31
- Error contribution 5-26
- Error messages 18-3
- Euclidean distance 5-2, 6-3
- Excel file 14-5
- Exclude 13-20, 16-15
- Exclusion sets 11-9, 12-32
- Exploratory Data Analysis
 - Example 2-7
 - Preparation 4-2

- F**
- F test 5-22
- Factor Selection 5-32
- FAQ 18-2

- Farthest neighbor clustering 5-5
- Feasible Region 8-21
- Feature selection 11-11
- File
 - Drag and drop 14-5
 - Format
 - Agilent ChemStation 14-10
 - AIA chromatography standard 4-23, 14-10, 15-5
 - Analect 14-11
 - ASCII 1-2, 14-5
 - ASD Indico Pro 14-10
 - Brimrose AOTF 14-11
 - Bruker OPUS 14-13
 - Excel 1-2, 14-5
 - Guided Wave 14-11
 - Hamilton Sundstram PIONIR 14-13
 - Hewlett Packard 8452 14-10, 14-12
 - JCAMP-DX 14-12
 - LT Industries 14-12
 - Perkin-Elmer spectroscopy 14-12
 - Pirouette 1-2, 14-5
 - Menu 16-3
 - Merging 16-5
- Fill
 - by PCA 13-18
 - for Mask 13-17
 - Options 13-13, 16-19
- Find 11-5
- Fisher weight 11-12
- Flexible link 5-3, 5-8
- Format
 - Galactic GRAMS 15-3
- French 10-22
- Frequently asked questions 18-2
- FUEL 9-3
- G**
- German 10-22
- Global scope (SIMCA) 16-24
- Go to 13-3
- Graphics
 - Capture of 15-2
 - Color 12-34
 - Creating subsets 12-32
 - Labeling 12-16, 16-35
 - Linking 12-28
 - Magnifying 12-8
 - Plot types
 - Line plots 12-13
 - Scatter plots 12-5 to 12-13
 - Scaling 12-11
 - Types 12-4
- Grid lines 10-9
- Group average link 5-3, 5-9
- H**
- Help
 - Menu 16-45
 - System description 1-6
- Hierarchical classification 6-26
- Hierarchical Cluster Analysis (HCA)
 - Activate class 12-27
 - Definition 5-1
 - Options 16-21
- Highlighting 10-1, 13-5
- Hotelling's T2 5-25
- HYDROCRB 9-3
- Hyperbox (for PCA and SIMCA) 5-30
- I**
- ID Tool 16-36
- Identity matrix 17-2
- Ill-conditioned matrix 18-7
- Incremental link 5-3, 5-9, 5-12
- Independent variable 4-5, 7-3, 13-5
- Indicator function 5-22
- Inner bound 8-23
- Insert 13-9
- Insertion cursor 10-6
- InStep 6-26, 9-6
- Interaction tools 16-36
- Interpolated fill 16-19
- Inverse (of a matrix) 17-6
- Inverse least squares 7-3
- Italian 10-22
- J**
- Jaggedness 7-9
- Japanese 10-22
- JCAMP file format 14-12
- K**
- Kennard-Stone 11-10
- K-Nearest Neighbors (KNN)
 - Definition 6-2
 - Misclassification 6-8
 - Model 6-5
 - Optimization 6-10
 - Options 16-23
- Kovats retention index 4-22
- L**
- Labels
 - Attributes of 10-17
 - Line plot axis 10-13
- Lack of Fit (in ALS) 8-5
- Language 10-21
- Latent variable 5-14
- Leverage 7-11, 7-12
- Limits 12-16
- Linear Learning Machine 6-1
- Linking
 - Complete link 5-7
 - HCA methods 5-3
 - in HCA 16-21
 - Single link 5-6, 5-11
 - Views 12-28, 13-6
- Loading data 14-3
- Loadings 5-18, 5-36, 7-18
- Local scope (SIMCA) 16-24
- Locally Weighted Regression 7-58
- Logarithm 4-13

- M**
- Magnify tool 12-8, 16-36
- Mahalanobis distance 5-25, 6-15, 7-48
- Marker, for alignment 4-22
- Mask
 - Preparing via Fill 13-17
 - to Indicate transfer samples 6-29, 7-56
 - Using 4-11
- Matrix 4-36, 17-2
- MCR
 - Discussion 8-12
 - Math 8-15
 - Options 8-4, 8-19
- Mean center 4-26
- Mean fill 16-19
- Median link 5-3, 5-8
- Menu
 - Display 16-34
 - Edit 16-11
 - File 16-3
 - Help 16-45
 - Objects 16-38
 - Process 16-19
 - Windows 16-39
- Merge
 - Description 16-5
 - Drag and drop 14-5
- Messages 18-3
- Metafile, EMF 15-2, 16-14
- Misclassification matrix 6-8, 6-24, 6-27
- Missing value
 - Finding 13-13
 - symbol in file (M) 14-8
 - symbol in table (*) 13-8, 13-13, 16-6
- Mixture analysis 8-1
- MNAPS 8-22, 9-4
- Model
 - ASCII 15-9
 - Galactic CAL file 15-8
 - Guided Wave calibration file 15-8
 - KNN 6-5
 - PCA 5-43
 - Regression 7-14, 7-49
 - Save 15-6
 - SIMCA 6-19
- Model files
 - PMF 16-7
- Model optimization
 - PLS 7-16
 - PLS-DA 7-38
 - SIMCA 6-25
- Modeling Power 5-27
- Momentum spinning 12-10
- Most recent files 16-11
- Mouse actions
 - Click-drag 10-1, 13-4
 - Control-click 10-2
 - Right mouse button 11-5, 12-2, 12-8
 - Shift-click 10-1
- MSC 4-21
- Multiple Linear Regression 7-3
- Multiplication 4-14
- Multiplicative Scatter Correction 4-21
- Multiplot 10-15, 12-20
- Multivariate Curve Resolution 8-12
- MYCALIGN 4-24, 9-4
- MYCOSING 9-4
- N**
- Names
 - in Plots 16-35
 - in Spreadsheet 13-1
 - of Objects 11-3
 - of Sets 11-6
 - of Windows 12-3
- Near infrared spectroscopy (NIR) 3-2
- Nearest neighbor 6-3
- Nearest neighbor clustering 5-5
- New 16-3
- NIPALS 5-28, 7-5
- Node of dendrogram 12-22
- Non-negativity 8-16
- Normalization
 - Examples 4-16
 - Maximum value 4-15
 - using a Mask 4-16
 - Vector area 4-14
 - Vector length 4-15
 - Vector range 4-15
- Notes 10-15, 11-4, 16-14
- Number of factors 5-19 to 5-27, 6-25, 7-6 to 7-9
- O**
- Object Manager
 - Creating charts from 11-8
 - Description 11-1
 - Finding text 11-5
 - Icons 11-1
 - Information 11-4
 - Naming subsets 11-6
- Objects
 - Deleting 11-7
 - Menu 16-38
 - Names 11-3
- OCT_TEST 3-29
- OCTANE20 3-4, 5-11, 9-4
- Optimization
 - F test 5-22
 - IND function 5-22
 - KNN 6-10
 - Number of factors 5-19 to 5-27, 6-25, 7-6 to 7-9
 - Number of neighbors 6-8
 - PRESS 7-7, 7-47
 - Regression models 3-20
 - SIMCA 6-25
- Options (setting of) 4-32
- Orthogonal Leverage 11-11
- Orthogonal signal correction (OSC) 7-12
- Outlier detection
 - Importance 5-23
 - in CLS 7-47
 - in Scatter plots 4-8
 - using Leverage 3-20, 7-11

- using Mahalanobis distance 5-38, 6-15
 - using Q statistic 5-25
 - using Sample Residual 7-48
 - using Studentized residuals 7-24
- Overview region (of dendrogram) 12-24
- P**
- PALMCHRO 14-3
- Panning 12-16
- Pareto scale 4-31
- Parsimonious model 5-27, 7-9
- Partial Least Squares (PLS)
 - Example 3-16
 - for Classification 7-38
 - Math 7-5
 - Model 7-14, 7-48
 - Optimization 7-16
 - Options 16-24
 - Prediction 3-29, 7-31
- PCA (see Principal Components Analysis)
- PCA Fill 13-18
- PCA Hypergrid 11-11
- PCR (see Principal Components Regression)
- Piecewise Direct Standardization 4-35
- PIONIR 14-13
- PIR file 1-2, 14-5, 15-3
- Plot
 - Labels 10-17
 - Preferences 10-16
 - Scaling 12-11
 - Symbols 10-11
- PLS (see Partial Least Squares)
- PLS-DA 7-38
- Plus sign cursor 10-6
- PMF 16-7
- Point
 - Default label 10-16
 - Labels 16-35
 - Size 10-11
- Pointer tool 10-2, 16-36
- Portuguese 10-22
- Pound sign (#)
 - ASCII file specifier 14-6
 - Examples 14-6 to 14-9
- Prediction
 - KNN 6-12
 - Options 10-19
 - PCA 5-43
 - PLS/PCR 7-31
 - Preferences 10-19
 - SIMCA 6-26
- Preferences
 - Color
 - Color sequence 10-18
 - General settings 10-7
 - Info Box Font 10-20
 - Language 10-21
 - Prediction 10-19
 - Sets of 10-21
 - Text 10-8
 - Views 10-7
- Preprocessing
 - Options 16-21
- PRESS 7-7, 7-47
- Principal Component Analysis (PCA)
 - Definition 5-13
 - Math 5-16
 - Options 16-22, 16-23, 16-24
 - Prediction 5-43
 - Terminology 5-14
- Principal Component Regression (PCR)
 - Math 7-4
 - Model 7-14, 7-48
 - Optimization 7-16
 - Options 16-24
 - Prediction 7-31
- Print setup 15-1, 16-10
- Printing 16-9
- Probability 5-23, 5-24, 7-22, 10-19
- Process Menu 16-19
- Projections
 - in SIMCA 6-24
- Pseudo-eigenvalue 7-5
- Pure Component Spectra 7-46
- Q**
- Q statistic 5-24
- Qualify
 - in KNN 6-9
 - in SIMCA 6-23
- R**
- RANDOM 5-4, 9-5
- Random sample selection 11-11
- Range scale 4-30
- Range tool 12-18, 16-36
- Rank 4-35, 5-16, 18-7
- Recent files 16-11
- Redraw 10-13
- Regression
 - Example 3-17
 - Linear 7-3
 - Model 7-14, 7-49
 - Multivariate 7-3
 - Prediction 7-31
 - Validation 7-7
 - Vector 7-24, 15-9
- Rename 11-6
- Residuals
 - Between classes (SIMCA) 6-20
 - Sample, in CLS 7-47
 - Sample, in PCA 5-23
 - Studentized 7-11
 - X-block, in PCA 5-38
- Results
 - Display 10-17
- Retention index 4-22
- Ribbon
 - Description 10-3
 - Edit tools 10-4
 - File and window tools 10-3
 - Navigation aids 10-5
 - Spin control 10-5

- View buttons 10-4
- Rotation
 - Spinning 3D views 12-9
 - Varimax 5-41
- Run status 18-5
- S**
- Sample residual 5-23, 7-47
- Sample selection 11-10
- Save
 - Data 16-5
 - Model 15-6
 - Objects 15-5
- Savitzky-Golay 4-12
- Scaling
 - by Range 4-30
 - by Variance 4-27
 - in Plots 12-11
 - Variables 4-17
- Scatter plots 12-5 to 12-13
- Scores
 - contributions to 5-26
 - Graphical description 5-17
 - in PCA 5-36
 - in Regression 7-17
- Screen capture to printer 15-1
- Scripting 19-1
 - Examples 19-8
- Scroll tools 13-4
- SEC 7-12
- Selecting data
 - Cloaking 12-8
 - in Charts 12-5
 - in Tables 13-5
- Selectivity 6-28
- Selector button 10-5, 12-14, 12-19
- Sensitivity 6-28
- SEVEN 9-5
- Shift-click 10-1, 16-5
- SIMCA (see Soft Independent Modeling of Class Analogy)
- Similarity 5-2, 12-23, 12-25
- Simplicity 5-28
- Single link 5-3, 5-6, 5-11
- Singular Value Decomposition (SVD) 7-4
- SMCR 8-12
- Smoothing 4-12
- SNV 4-22
- Soft Independent Modeling of Class Analogy (SIMCA)
 - Definition 6-15
 - Model 6-19
 - Optimization 6-25
 - Options 16-24
 - Prediction 6-26
- Sort 13-11
- Source
 - Amounts 8-20
 - Apportionment 8-1, 8-24
 - Profiles 8-20
- Spanish 10-22
- SPE 5-24
- Spin control buttons 12-10
- Spinner Tool 12-9, 16-36
- Spreadsheet
 - Cursors 10-6
 - Entering data 14-1
 - Labels
 - in ASCII file 14-6
 - in spreadsheet file 14-9
 - Navigation 13-2
 - Variable types 13-10
- Squared prediction error (SPE) 5-24
- Standard deviation 5-13
- Standard Error
 - of Calibration (SEC) 7-7, 7-12, 7-47, 7-49
 - of Prediction (SEP) 7-7, 7-47
 - of Validation (SEV) 7-8
- Standard Normal Variate 4-22
- Statistical Prediction Error (SPE) 7-46
- Studentized residual 7-11
- Submenu 16-1
- Subset selection 4-34
- Subsets
 - by Variable selection 11-11
 - from Plots 12-32
 - from Sample selection 11-10
 - from Spreadsheets 13-20
 - making a Full Data set 11-9, 18-2
 - Naming 11-6
 - Removing 11-7
- Subtraction 4-18
- Support 18-14
- Symmetric matrix 17-2
- T**
- Technical assistance 1-6, 18-14
- TERNARY 9-5
- Test set 3-29
- TIFF 15-2
- Total contribution 5-26
- Total Modeling Power 6-21
- Training set 4-5
- Transfer of calibration 4-33, 4-34, 6-29, 7-56
- Transfer samples 6-29, 7-56
- Transforms 4-10, 16-28
- Transmittance spectra 4-13
- Transpose
 - function 13-12, 16-7
 - of a matrix 17-3
 - of data 14-7
- Troubleshooting 18-1
- U**
- Uncertainty 7-12
- Undo 16-13
- Unmagnify 12-8
- Unzoom 10-4
- User charts 12-1
- User permissions 1-5, 18-1
- V**
- Validation

- Regression 7-7
- Standard Error of 7-8
- Variable
 - Dependent vs. independent 4-5
 - Selection 11-11
 - Types 13-10
- Variance 5-13, 5-23, 7-47
- Variance scale 4-27
- Variance weight 11-12
- Varimax 5-28 to 5-29, 5-41
- Vector
 - Area normalization 4-14
 - Definition 17-1
 - Length normalization 4-15
- Version of software 16-46
- View
 - Preferences 10-7
 - Types 10-4, 12-4
- W**
- Warning messages 18-8
- Web site 18-14
- Weight loadings 7-6
- Window
 - Preferences 10-16
- Titles 12-3
- Window size
 - Align transform 4-23
 - PDS 4-35
- Windows Explorer, load data from 14-4
- Windows Menu 16-39
- X**
- X Limits 12-16
- X Preprocessed 5-32
- X Residuals
 - in PCA 5-38
 - in PLS 7-26
 - Prediction 7-33
- X variable 4-5, 13-5
- XLS file 1-2
- XRF 9-5
- Y**
- Y variable 4-5, 13-5
- Z**
- Zero fill 13-13, 16-19
- Zoom button 10-4, 12-20