# Contents

# `ggobi` Manual

Deborah F. Swayne, AT&T Labs – Research
Di Cook, Iowa State University
Andreas Buja, AT&T Labs – Research
Duncan Temple Lang, Lucent Bell Labs

February 2002

### Abstract

The GGobi software is a data visualization system with state-of-the-art interactive and dynamic methods for the manipulation of views of data. It represents a significant improvement on its predecessor, XGobi, with multiple plotting windows, more flexible color management, XML file handling, and better portability to Windows.

The most significant change may be that ggobi is so easy to extend, either with the addition of "plugins" or by embedding it in other software; either way, it can be controlled using an API (application programming interface). When ggobi is embedded in R, for example, the result is a full marriage between ggobi's direct manipulation graphical environment and R's familiar extensible environment for statistical data analysis.

It has the same graphical functionality whether it is running standalone or embedded in other software. That functionality includes 2-D displays of projections of points and edges in high-dimensional spaces, as well as scatterplot matrices, parallel coordinate, time series plots and bar charts. Projection tools include average shifted histograms of single variables, plots of pairs of variables, and grand tours of multiple variables. Views of the data can be reshaped. Points can be labeled and brushed with glyphs and colors. Several displays can be open simultaneously and linked for labeling and brushing. Missing data are accommodated and their patterns can be examined.

# 1 Introduction

This paper gives an overview of the layout and functionality of GGobi, interactive graphical software for exploratory data analysis. Readers who are familiar with XGobi will find much that is familiar in GGobi's design, and might want to read section 11 first, where key differences between the two programs are described. There are several papers that describe parts of the functionality existing in both packages [6, 14, 13, 4, 8, 9, 7, 5].

You will find that you can use GGobi for simple tasks with virtually no instruction. All that a user needs is some cursory knowledge of the developments in interactive statistical graphics of the last 15 years, as well as a willingness to experiment with the sample data provided and guided by the tooltips. In parallel with the hands-on learning process, it is probably useful to acquire a basic understanding of the overall layout and functionality of the system. The greatest success is obtained by users who have gained experience with the system and combine it with creativity and data analytic sophistication.

We begin with a tutorial, and move on to describe GGobi in detail.

# 2 Tutorial

Several sample data files are included with the GGobi distribution, in a directory called **data,** and there you will find the file **olive.xml**, a dataset on olive oil samples from Italy [10]. This data set takes advantage of an XML input/output format, a feature in GGobi that wasn't available in XGobi. The olive oil data consists of the percentage composition of 8 fatty acids (palmitic, palmitoleic, stearic, oleic, linoleic, linolenic, arachidic, eicosenoic) found in the lipid fraction of 572 Italian olive oils. There are 9 collection areas, 4 from southern Italy (North and South Apulia, Calabria, Sicily), two from Sardinia (Inland and Coastal) and 3 from northern Italy (Umbria, East and West Liguria).

The data is part of a quality control study of olive oils. It is proposed that the olive oils from different regions have different fatty acid signatures.

Start GGobi for the olive oils data by typing:

```
ggobi olive
```

Two windows will appear, the GGobi console and a scatterplot, as shown in Figure 1.

The console has a panel of controls on the left, labeled **XYPlot**, and a variable selection region on the right. You can see that the scatterplot contains a 2-dimensional projection of the data, a plot of Area vs Region. Move the mouse to one of the variable labels in the variable selection region, and leave it in place until the tooltip appears, explaining how to select new variables for the plot. Begin to get a feeling for the data by looking at several of the 2d plots: Area vs palmitic, Region vs oleic, etc.

Next, get acquainted with the main menubar for the console by exploring each of its menus. Pay particular attention to the Display, ViewMode and Tools menus.

- The Display menu is the interface for opening new plotting windows.

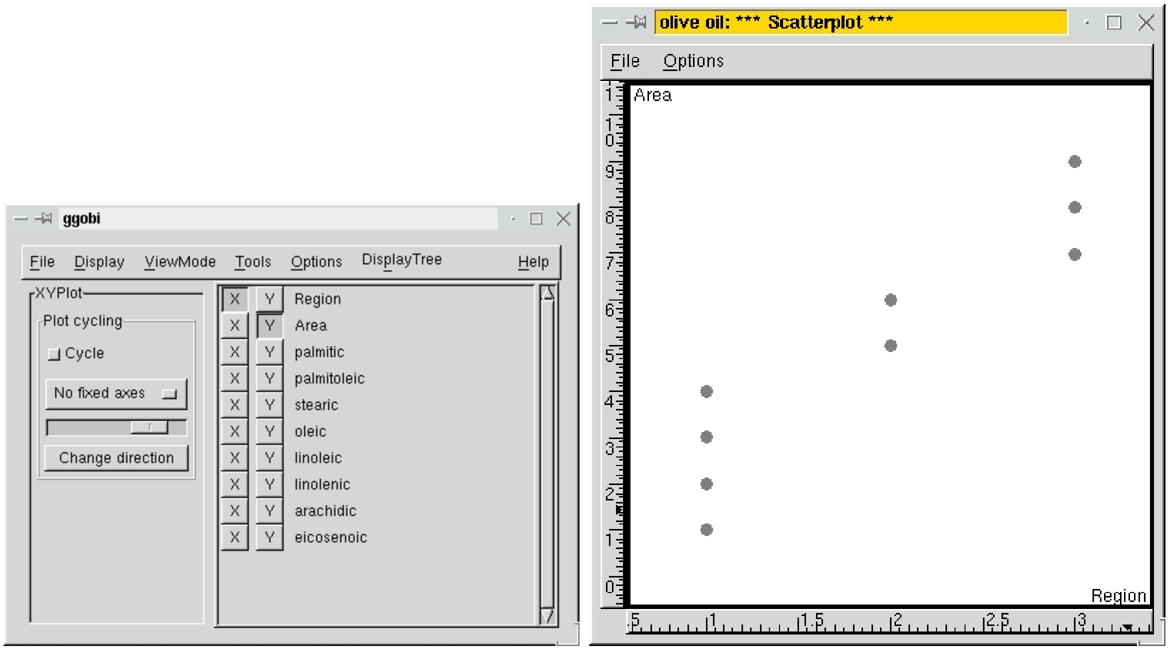


Figure 1: Layout of a ggobi session. The plotting window contains a scatterplot of the Area vs Region, from the olive oils data.

- The ViewMode menu is the interface for specifying both the projection (1d, 2d, 3d or higher) and mouse interactions (for scaling the plot, highlighting points, and so on) for the current plot.
- The Tools menu lets you open other windows to manipulate characteristics of the data and the view.

Using the ViewMode menu, choose **Identify**. Look at the buttons inside the leftmost portion of the ggobi console. Notice that they're contained by a frame labeled **Identify,** which is the view mode of the current plot. This frame contains most of the row labeling controls, which are described in section 6.9. Move the cursor around the plot window using the mouse, and observe the labels of the point closest to the cursor. The labels for this data set show the geographic area where the sample was taken.

Using the Display menu, open a barchart display. Notice that the new window has a narrow white band drawn around the outside of the plotting area: that means that the new window is now the "current display" and its plot is the "current plot." Click in the plotting region of the other scatterplot to make it the current plot, and notice what happens in the console when you alternate between the two. Both the controls and the variable selection region correspond to the current plot.

Now set up a plot of linoleic vs eicosenoic in the first scatterplot, and disply Region in the barchart, and make the barchart the current plot. Using the ViewMode menu, choose **Brush.** Look at the buttons and menus inside the leftmost portion of the ggobi console. Notice that they're contained by a frame labeled **Brush,** which is the view mode of the current plot. This frame contains most of the brushing controls, which are described in the section 6.8. A few of the brushing controls, though, are in the main menubar: in the Reset menu and at the bottom of the Options menu.

The rectangle that appears in the current plot is the "paintbrush," and dragging it over bars (or in a scatterplot the points) changes their color. Change the color of the brush by opening up the **Choose color & glyph** panel (Figure 2). Hold down the left button and drag the mouse to paint the first region, then the second and third regions, or click on the bars. Since you're brushing in the **Transient** style, the points resume their original color when the paintbrush no longer surrounds them.

While you brush, keep an eye on the plot of linoleic vs eicosenoic, and notice where the painted points fall in that scatterplot (Figure 3). The oils from region 1, south Italy, are separable from the other two regions using eicosenoic acid. Using two linked 2d plots is one way to explore the relationship among three variables.

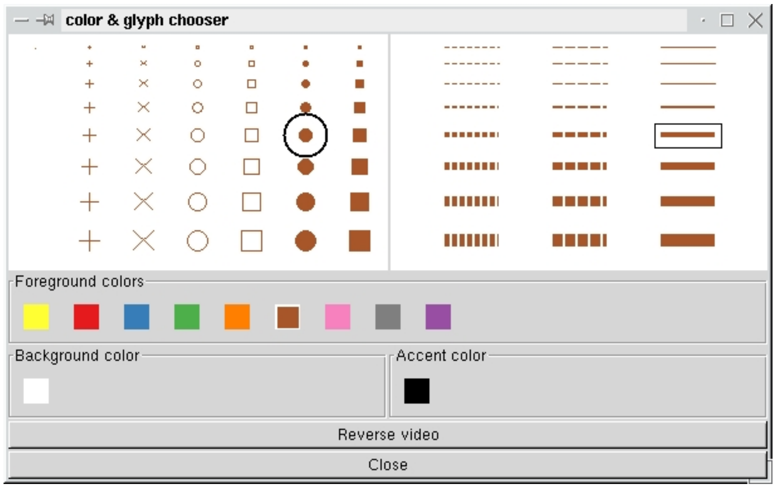


Figure 2: The chooser for selecting color and glyph for painting points, as well as type and thickness for painting lines. It also allows setting the plot background color, and has a full color wheel for tuning the colors.

Change to **Persistent** brushing and paint the three Regions using different colors.

Open the **Variable transformation** tool, select all of the fatty acid variables – either hold down the control button while selecting them one at a time, or select the first one, then hold down the shift button down while you select the last one. Once they're all highlighted, choose **Standardize** in the "Stage 2" transformation panel to standardize all the variables to have mean 0 and variance 1. This tool has numerous transformations, which can be applied in sequence. Stage 0 transformations are used to adjust variables' values so that they're within the range of some stage 1 transformations. Stage 2 transformations allow for post-processing of stage 1 transformations, so that the variables can be logged and then standardized, for example.

But that was just an instructive detour; click "Reset all" to to turn off standardization and any other transformations you've explored.

Now look at the plot of linoleic vs eicosenoic. Open up the **Color & glyph groups ...** tool, and experiment with it: Use the "Shadow" toggle buttons to have groups of points drawn in a dim color, and then use "Exclude shadows" to exclude them altogether, and "Include shadows" to re-include them. (Figure 5). Finally, use these controls to shadow brush the cases from geographic region 1, and then use **Exclude shadows** to remove them from consideration.

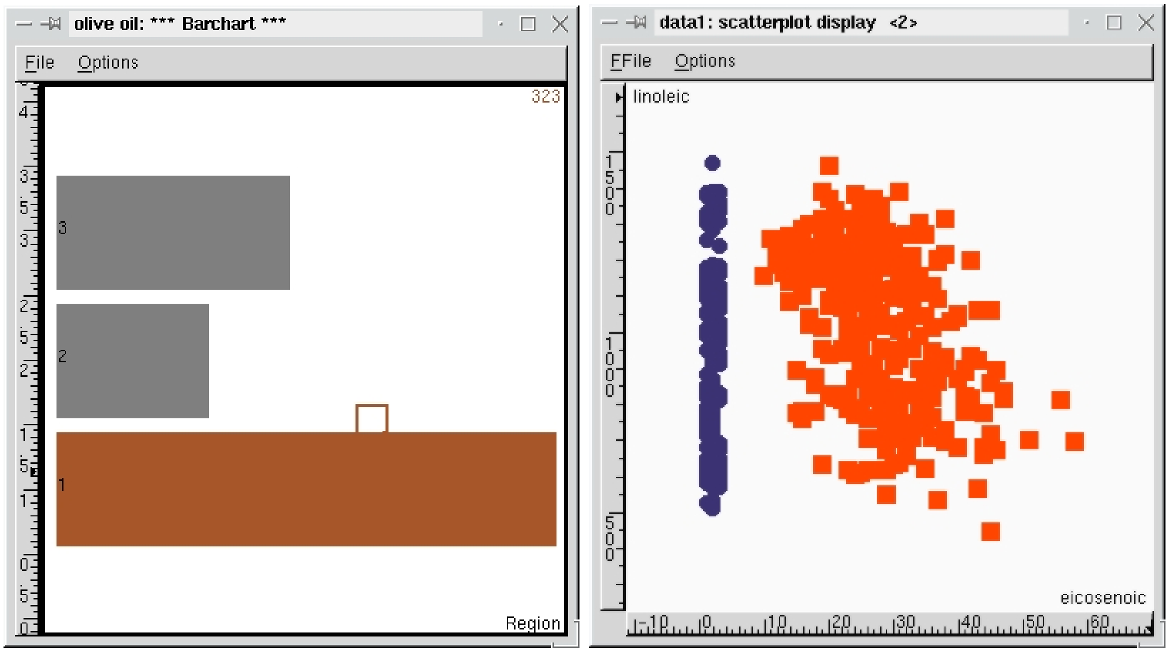
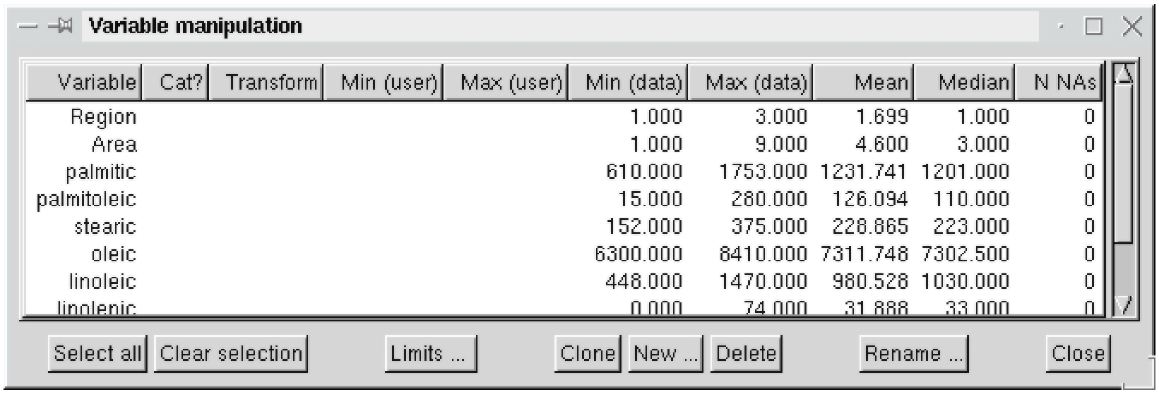
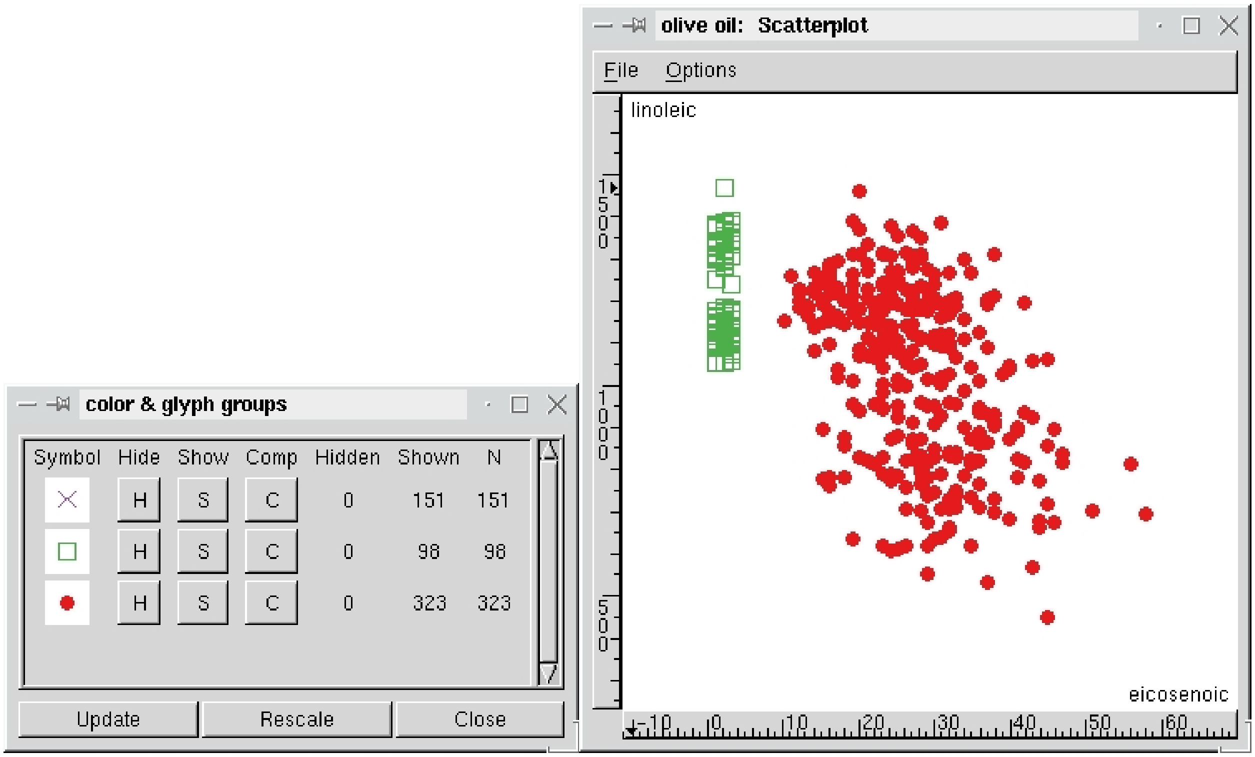


Figure 3: Brushing in a GGobi session. Brushing Region 1 shows that it corresponds to a cohesive cluster in the variables linoleic and eicosenoic acid.



| Variable | Cat? | Transform | Min (user) | Max (user) | Min (data) | Max (data) | Mean | Median | N NAs |
|---|---|---|---|---|---|---|---|---|---|
| Region | | | | | 1.000 | 3.000 | 1.699 | 1.000 | 0 |
| Area | | | | | 1.000 | 9.000 | 4.600 | 3.000 | 0 |
| palmitic | | | | | 610.000 | 1753.000 | 1231.741 | 1201.000 | 0 |
| palmitoleic | | | | | 15.000 | 280.000 | 126.094 | 110.000 | 0 |
| stearic | | | | | 152.000 | 375.000 | 228.865 | 223.000 | 0 |
| oleic | | | | | 6300.000 | 8410.000 | 7311.748 | 7302.500 | 0 |
| linoleic | | | | | 448.000 | 1470.000 | 980.528 | 1030.000 | 0 |
| linolenic | | | | | 0.000 | 74.000 | 31.888 | 33.000 | 0 |

Figure 4: The variable manipulation panel contains basic summary statistics of variables. It can also be used for adding new variables and for variable subset selection before launching multi-plot displays.



| Symbol | Hide | Show | Comp | Hidden | Shown | N |
|---|---|---|---|---|---|---|
| × | H | S | C | 0 | 151 | 151 |
| □ | H | S | C | 0 | 98 | 98 |
| ● | H | S | C | 0 | 323 | 323 |

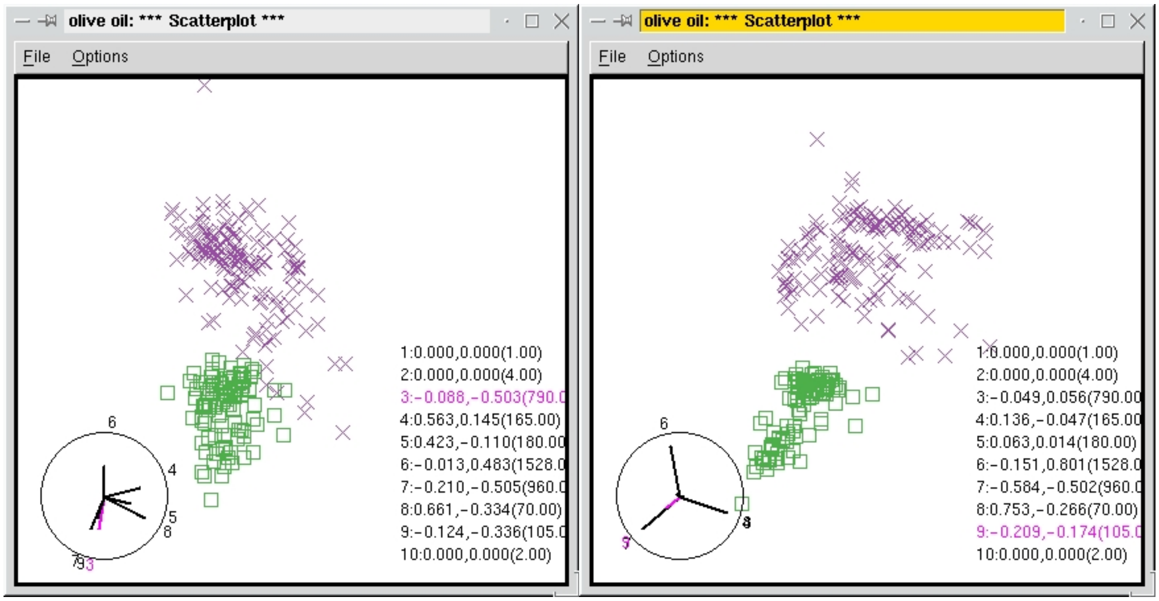
Figure 5: Toggling clusters of points on and off according to color/glyph value.

Switch into **2D Tour** using the **ViewMode** menu. A column of variable circles is added to the right of the variable toggles, one for each selected variable. Click on the toggle widgets for the variables palmitoleic through arachidic to make these variables available for touring, and toggle Region and Area out of the tour. Click the **Reinit** button. The tour should now include 7 variables, with one variable circle for each.

The scrollbar at the top of the tour controls is used to adjust the speed of the tour. Drag it to the right to speed up the rotation. The circle at the bottom left of the plot window displays the axes for the tour. These can be removed by toggling the "Show axes" button on the plot window's **Options** menu. Pause the tour when you see a separation of the two regions (as seen in Figure 6).

Now you are going to use manual tour controls to sharpen up the separation by manually changing the coefficients of the variables in the projection. Click on the **Manip** (magenta) button below the variable selection region of the console, and then click on linoleic acid (a magenta circle will be seen now in the variable circle for linoleic acid). **Oblique** is already selected in the **Manual manipulation** menu. In the plot window, hold down any mouse button while you drag the cursor. The coefficient corresponding to linoleic acid will increase and decrease following the mouse, inducing a rotation of the scattercloud. If the axes are still showing in the scatterplot window, the axis in magenta corresponds to the variable you're manipulating. Similarly rotate oleic acid and arachidic acid in and out of the plot, with the aim of finding a projection where the two regions are well separated (for example, Figure 6). There are 5 choices of manipulation mode (unconstrained oblique, vertical, horizontal, radial and angular) to explore.

Next use the Tools menu to open the **Variable manipulation** tool (Figure 4). It contains a notebook widget which separates the variables by type, and displays information about them in a


olive oil: *** Scatterplot ***
File
Options
6
4
5
8
7
9
3
1
olive oil: *** Scatterplot ***
File
Options
6
7
8

```
1:0.000,0.000(1.00)
2:0.000,0.000(4.00)
3:-0.088,-0.503(790.0
4:0.563,0.145(165.00)
5:0.423,-0.110(180.00
6:-0.013,0.483(1528.0
7:-0.210,-0.505(960.0
8:0.661,-0.334(70.00)
9:-0.124,-0.336(105.0
10:0.000,0.000(2.00)
```

```
1:0.000,0.000(1.00)
2:0.000,0.000(4.00)
3:-0.049,0.056(790.00
4:0.136,-0.047(165.00
5:0.063,0.014(180.00)
6:-0.151,0.801(1528.0
7:-0.584,-0.502(960.0
8:0.753,-0.266(70.00)
9:-0.209,-0.174(105.0
10:0.000,0.000(2.00)
```

Figure 6: Tour plots revealing difference between regions 2 and 3 of the olive oils.

set of tables. The buttons below the tables allow you to set variable limits, add new variables, and a few other things. Try clicking the mouse in the table – you can select one row of the table at a time, or use the control and shift keys as modifiers to select more than one. Select all the fatty acid variables, all of which are in the table of real variables.

Open a parallel coordinates display using the **Display** menu. Select the first plot in the parallel coordinates display by clicking on it, and use the **ViewMode** menu to switch to **Brush** mode. Choose a new color (say yellow) and large closed glyph, and transiently paint the case with a very low value on palmitic (Figure 7). This case also has a very high value for oleic acid, and low value for linolenic acid.

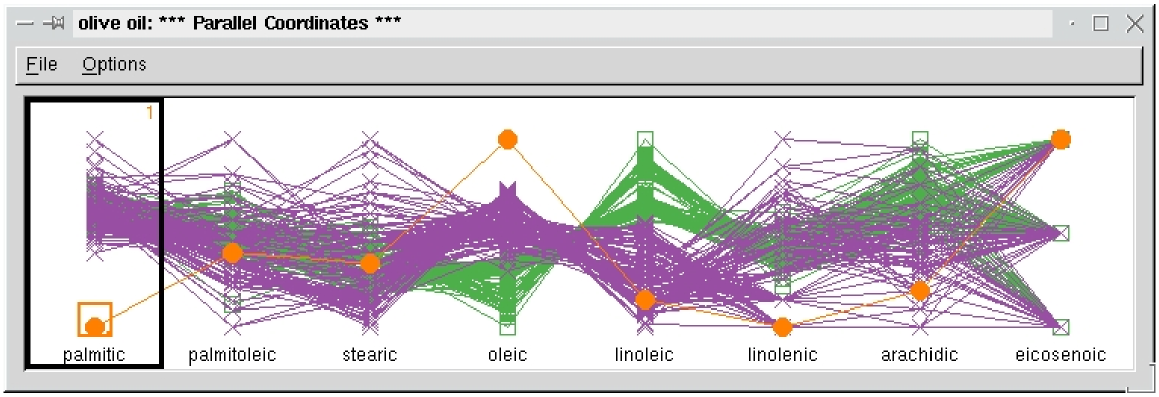


Figure 7: Brushing in a parallel coordinates display reveals an outlier in palmitic, oleic and linolenic acids.

Using the **Color & glyph groups** tool from the **Brush** controls, click on the appropriate 'S' button to bring the Region 1 cases back into the plot. Using the **File** menu, save the data, preserving the colors and glyphs that have been assigned during this session.

This has been a brief introduction to the use of GGobi. The following section contains more detailed information on its functionality.

# 3 Layout and functionality

## 3.1 The major functions

Across the top of the console, as seen in in Figure 1, stretches a row of menu buttons: **File**, **Display,** **ViewMode**, **Tools**, **Options** and **DisplayTree** are always visible; others appear as appropriate.

As expected, the **File** menu contains items for selecting input/output functions and for exiting.

The **Display** menu allows a new plotting window to be opened. The display types include

- scatterplot,
- scatterplot matrix,
- parallel coordinates plot,
- time series plots, and
- barchart.

Each display type is discussed in section 3.2.2.

The **ViewMode** menu contains items to set the projection and to the interaction type:

- 1DPlot: 1-D dotplots and average shifted histograms,
- XYPlot: 2-D scatterplots,
- 1D Tour: 1-D tour,
- Rotation: 2-D tour constrained to use exactly 3 variables,
- 2D Tour: 2-D tour,
- 2x1D Tour: a correlation tour; that is, independent 1-D tours on horizontal and vertical axes,
- Scale: axis scaling,
- Brush: setting point glyphs, edge types, and point and edge colors,
- Identify: labeling points,
- Edit Edges: add points or edges.
- Move Points: direct manipulation of point positions,

When you choose a new view mode, the controls at the left of the main window will change correspondingly: each mode has its own parameters, and its own rules for responding to mouse actions in the plotting windows. The view modes are discussed in section 6.

The **Tools** menu gives access to

- a variable manipulation table,

- a variable transformation pipeline,
- a variable sphering panel,
- jittering controls,
- a panel for selecting a new color scheme for drawing, and then for automatically brushing points and edges by mapping the color scheme onto a selected varriable.
- a panel for brushing and excluding groups of cases,
- subsetting functions for systematic and random subsampling,
- a tool for managing missing values.

Each of these tools is discussed in section 7.

There are a couple of distinctions between view modes and tools. The view mode functions determine the mouse interactions in the display windows, while none of the Tools does. Furthermore, view mode functions populate the control panel in the ggobi console, while tools open their own windows, as shown in Figure 4. For example, the **Variable Manipulation tool** is a window containing tables which describe the variables, as well as several buttons, and the **Variable Transformation tool** is also a separate window, with a list of variable names and a set of transformation menus.

The **Options** menu allows users to set some options for the main control window: whether tooltips are displayed and whether the control panel is shown. In some view modes, it contains additional options that are specific to that mode.

Other menus are present only during certain modes: you will sometimes find an **I/O** menu or a **Reset** menu.

The **DisplayTree** menu allows users to open a tree listing all the currently open display windows, each of which may contain several plots.

## 3.2 Graphical displays

### 3.2.1 Current display, current plot

Since there are multiple displays, some of which contain multiple plots, the question arises: Which plot in which display window corresponds to the console? If you select the **Scale** mode, how can you tell which plot is going to respond?

There is in GGobi a notion of the "current display" and the "current plot." (We need both because some displays, like the scatterplot matrix, contain multiple plots.) The current plot is the one which is outlined with a thick white border; the current display is the one which contains the current plot.

To reset the current plot and display, just click once (left, right or middle) in the plot you wish to address. To understand the effects of this selection, open a few displays and set them in different ViewModes, then click on different plots and see what happens. The white border follows your actions, and the console updates so that its control panel corresponds to the current display type and view mode.

### 3.2.2 Display types

Each display type is briefly described here. As mentioned earlier, there are presently five main display types:

The **scatterplot** display is a window containing a single scatterplot. By default, it includes axes; they can be turned off using the Options menu in the main control window. It has the largest number of view modes of any display, and each of the projection modes has its own rules for variable selection. The main variable selection interface for the XYPlot modes uses toggle buttons: clicking on the $X$ button selects the variable to plot horizontally, and clicking on $Y$ selects the variable to plot vertically. (There's a less obvious variable selection interface as a shortcut: clicking left (right) on the variable label selects the $X$ ($Y$) variable.) The interface for the 1D plot is similar: it only shows one column of toggle buttons, so clicking on $X$ selects the variable to be plotted irrespective of the plot orientation. (Using the variable label shortcut, you can change the plot orientation as you make a variable selection.) The tour modes, including rotation, all use toggle buttons to select a subset of variables to be available for touring, and a column of variable circles, one for each variable in the subset. The variable circles can be used to further refine the selection of variables that are actively touring, and they provide some feedback about current projection. The variable selection behavior for the tour modes will be described in the section for each mode.

The **scatterplot matrix** is a window containing a symmetric matrix of scatterplots for the chosen variables. The plots along the diagonal are ASHes (Average Shifted Histograms). The matrix is required to be symmetric, and that constraint affects its variable selection behavior.

- Replace: First select one of the ASH plots along the diagonal to tell ggobi uniquely which variable to replace, then click on one of the toggle buttons in the variable selection region.
- Insert: First select one of the plots along the diagonal to tell ggobi uniquely where to insert the new plot, then click on one of the buttons. (GGobi will not add a variable that's already plotted.)
- Append: Click on one of the buttons to append a new plot after all other plots. (GGobi will not add a variable that's already plotted.)
- Delete: No plot selection is required; just select the variable you want to delete.

For the scatterplot matrix display, the variable label "shortcut" works, but it's simply redundant: that is, it makes no difference which button you press.

The **parallel coordinates** display contains a single parallel coordinates plot, which can be arranged horizontally or vertically. (To understand this plot if you are encountering it for the first time, imagine deconstructing a high-dimensional scatterplot and arranging its axes in parallel instead of orthogonally. To represent case $i$, think of drawing a dot on each axis, with the point on axis $j$ being the value of $x[i][j]$, and then connecting the dots into one set of connected line segments [11, 18].) The line segments are drawn by default, but you can turn them off using the **Options** menu on the display menubar.

By default, the plots are simple dotplots, but they can also be drawn using one of the two methods for 1D plots: as a textured dot strip or an ASH.

The variable selection behavior works as follows:

- Replace: First select one of the plots, then click on one of the toggle buttons to replace its plotted variable.
- Insert: First select one of the plots, then click on one of the toggle buttons to insert a new plot before the current plot. (GGobi will not add a variable that's already plotted.)
- Append: Click on one of the toggle buttons to append a new plot after all other plots. (GGobi will not add a variable that's already plotted.)
- Delete: No plot selection is required; just select the variable whose plot you want to delete.

For the parallel coordinates display, the variable label "shortcut" works, but it's simply redundant: that is, it makes no difference which button you press.

The **time series display** contains a row or column of 2-variable plots with a common axis, usually a time variable. By default, the points are connected with line segments.

The behavior of the toggle buttons and labels depends on the state of the **Selection mode** option menu in the console.

- Replace: If you want to replace the horizontal (time) variable, no plot selection is required; simply click the *X* toggle button for the variable you want. To replace a vertical variable, first select the plot you want to change, and then click the *Y* toggle for the variable you want. (For now, ggobi won't let you plot the same variable twice.)
- Insert: Select a plot in the display, and then click on a the *Y* toggle to add a new plot before the current plot.
- Append: Click on one of the *Y* toggles to append a new plot after all other plots. (GGobi will not add a variable that's already plotted.)
- Delete: No plot selection is required; just click the *Y* toggle for the variable in the plot you want to delete.

The variable label shortcuts can be used in the Time Series display. In general, clicking left (right) corresponds to selecting the *X* (*Y*) toggle button.

The **barchart display** contains a single plot, a barchart if the variable is categorical, and a histogram if it is real. Variable selection is simple: one variable at a time. An option menu can be used to switch to a spineplot display style, where all the bars are the same height, and it is their width that varies instead.

# 4 Data format

## 4.1 ASCII

The ascii data format used in XGobi is still supported, with some changes and some exclusions.

The only essential file is the one containing the data itself. Each line in the file contains one row of the input data matrix, and lines must be separated by carriage returns. Columns, or variables, can be separated by any number of tabs or spaces. The file needs to have the suffix *.dat*.

You can supply variable and case labels in associated files. Variable (column) labels can be in a file named **datafile.col** (or **datafile.column**, **datafile.collab**, or **datafile.var**). Variable ranges can be specified by adding two more fields, using | as the field separator. Case (row) labels can be supplied in a file named **datafile.row** (or **datafile.rowlab** or **datafile.case**). There should be one label per line, and the label can include blanks. If variable or case labels are not supplied, default labels using column or row numbers are used.

The files **datafile.glyphs** and **datafile.colors** can be used to set the plotting characters and colors to be used in drawing each point. Glyphs can be specified in two ways:

- with a string for glyph type and a number for glyph size, where the glyph type is one of
  - "." (a single-pixel point),
  - "plus",
  - "x",
  - "oc" (open circle),
  - "or" (open rectangle),
  - "fc" (filled circle), or
  - "fr" (filled rectangle).

  and the glyph size is between 0 and 7, or
- with one number per line, where the number is between 0 and 43. Here's how to generate that number: the type is between 0 and 6, using the ordering just presented, and the size is between 0 and 7. The number is then 0 for the single-pixel glyph, and $7 \times (type - 1) + size + 1$ for all other glyphs.

Filled circles may be the most visually appealing glyph, but take much longer to draw under Microsoft Windows. Consequently, our sample data files usually use a small filled rectangle as the default glyph.

Colors must be integers from 0 to one less than the number of colors in the colormap.

## 4.2 XML

The XML format is described in *Using XML Input Formats*, available from the web site. The XML format allows more detailed specification, such as

- multiple datasets within a single XML file, all available within a single GGobi process,
- rules for linking between datasets, and
- rules for specifying edges: line segments connecting pairs of points.

### 4.3 Database access

It is possible to interface ggobi with either a Postgres or MySQL database. Details of this are described in the document *GGobi and Database Management Systems*, available through the web site.

Database access is implemented as a plugin.

## 5 Integration of ggobi with other software systems

There are two ways to integrate ggobi with other software: by embedding ggobi in other software (that is, compiling ggobi into a shared object file and linking against it) or by extending ggobi with plugins. Either way, the application programming interface (API) makes it possible to control ggobi "from the outside."

The main examplea of embedding that currently exists is the integration of ggobi in R. This is described in the document *Using the R-GGobi Link* available through the web site. It will be possible to do the same thing from Perl and Python. There also exists a ggobi plugin for Gnumeric.

There are several examples of plugins for ggobi now distributed with the code, including a simple spreadsheet viewer.

## 6 View modes: projection and interaction

Selecting any mode on the **ViewMode** menu changes the interactions available for the display and plot that are current, and pops the corresponding control panel into the left portion of the console window. The modes in the top half of the menu do something more as well: they set the projection method for the display, and the meaning of actions in the variable selection panel always conforms to the current projection method.

As an example, start GGobi with some data, and watch what happens in the main GGobi window and a a single scatterplot display. When GGobi starts, it's in **XYPlot** mode by default, so the scatterplot window shows a 2-dimensional projection of the data. If you select **Scale** or **Brush** (or any choice in the bottom half of the menu), the control panel at the left of the console changes, reflecting the different interactions available to you in each mode. However, the projection in the window doesn't change. (There may be cues added to the plot to tell you the view mode of the plot.) If you click on the checkboxes in the variable selection panel at the right of the GGobi console, you replace one of the plotted variables.

Now select *2D Tour* in the ViewMode menu. *Everything* changes at once, because you've effectively selected both a new mode and a new projection type at the same time.

- The panel at the left of the console changes, because a new set of interactions just became available.
- The variable selection panel changes, because the variable selection behavior for high-dimensional projection types is quite different than that for low-dimensional projection types.

- The plot in the scatterplot display changes, because it's now showing a projection of 3 variables instead of 2. Furthermore, it's moving, because a grand tour process is running.

If you now select one of the modes in the bottom half of the ViewMode menu, you'll see again that the variable selection panel doesn't change, and the plot in the scatterplot doesn't change – except that it stops moving. The only thing that changes with every selection is the panel at the left of the console.

Now we'll describe each view mode in more detail, starting at the top of the ViewMode menu.

## 6.1 1D plots

The 1D plot can be displayed in two ways: as a textured dot plot or an average shifted histogram, or ASH.

The textured dot plot uses a method described in [17]. This method spreads the data laterally by amounts that are partly constrained and partly random, resulting in a fairly smooth spreading of the points and minimizing artifacts of the plotting method, such as stripes, clusters, or gaps.

The ASH is due to Scott ([12]), and the code is also his. In this method, several histograms are calculated using the same bin width but different origins, and the averaged results are plotted. His algorithm has two key parameters: the number of bins, which controls the bin width, and the number of histograms to be computed. In ggobi, the number of bins is held constant (at 200), while the smoothing parameter available on the console controls the number of histograms (which ranges from 1 to 50). The effect is a smoothed histogram − a histogram that allows us to retain case identity so that the plots can be linked case by case to other scatterplots.

Line segments can be added that run between the plotted point and the baseline of the ASH. This is helpful when the smoothing parameter is low, because it helps your eye make out the the shape of the ASH.

The 1D plot will be arranged horizontally if you select a variable with a left click, and vertically if you use a middle or right click.

The cycling controls can be used to make ggobi step through the plots automatically, one after another.

To activate this view mode from the keyboard, type *d* or *D* with the focus in the plot window, or type Control-*d* or Control-*D* with the focus in the console.

## 6.2 XY plots

The XY plots are the rudimentary 2 variable scatterplot (or draughtsman plot) displays. Two variables are chosen, one to be plotted horizontally and the other vertically. The cycling controls can be used to make ggobi step automatically from one pairwise plot to the next.

To activate this view mode from the keyboard, type *x* or *X* with the focus in the plot window, or type Control-*x* or Control-*X* with the focus in the console.

## 6.3 1D Tour

The 1D tour generates a continuous sequence of 1-D projections of the active variable space. The projected data are displayed as an average shifted histogram (ASH), horizontally or vertically. The scrollbar at the top of the controls allows the speed of rotation to be adjusted. The pause checkbox stops and starts the tour. **Reinit** initializes the tour to the projection (the ASH) of the first active variable. **Scramble** sets the view to a random projection.

Variables can be toggled into and out of the subset of active variables by clicking on the the toggle buttons. The variable will be immediately removed from the tour.

The variable circles on the right hand side of the control panel add further control for adding or removing variables from the current tour. The active variable space is the subset of variables currently selected, and their variable circles are drawn with a bold outline. When a variable is de-selected on the variable circles the variable fades out gradually, to maintain continuity of motion.

The reason for the two displays is to make handling of large numbers of variables more convenient. It might not make sense to include all the variables into a tour at once. The toggle buttons provide an efficient way to interact with the variable list. The variable circles provide information about the variables in the tour, how they are project to give the current projection, and information on which variable is the current manip variable. It is possible to select and de-select these variables to fine tune the tour.

The variable projection coefficients can be manually manipulated using manual controls. To select the variable to manipulate, click on the purple **Manip** button and then click on the variable circle. Horizontal mouse motions in the plot window then alter the coefficient for the manipulation variable, constrained by the values of the coefficients of other active variables (which may also change).

The variable axes and projection coeeficient values can be toggled on or off using the options menu at the top of the plot window.

To activate this view mode from the keyboard, type *t* or *T* with the focus in the plot window, or type Control-*t* or Control-*T* with the focus in the console.

### 6.3.1 Projection Pursuit

A guided tour is available when the **Projection Pursuit** button is selected. It is controlled through a separate pop-up projection pursuit window, which contains a plot of the projection pursuit index. When **Optimize** is selected, the tour is guided by the index rather than proceeding randomly. The numbers displayed to the right of the **PP index** label are the minimum, current value, and maximum of the index. A selection of indices is available.

### 6.3.2 Options

The **Options** menu on the main menu bar contains controls for laying out variable circles in the variable selection panel in different ways, and also for toggling variable fading on or off. Variable fading means a variable smoothly fades out when it is de-selected. The alternative is to zero the variable out of view immediately, which creates a discontinuity in the tour motion, but is desirable for some situations.

## 6.4 2D Tour

The 2D tour generates a continuous sequence of 2-D projections of the active variable space. The projected data are displayed as a scatterplot. The scrollbar at the top of the controls allows the speed of rotation to be adjusted. The pause checkbox stops and starts the tour. **Reinit** initializes the tour to the projection of the first two active variables. **Scramble** sets the view to a random projection.

Variables can be toggled into and out of the subset of active variables by clicking on the the toggle buttons. The variable will be immediately removed from the tour.

The variable circles on the right hand side of the control panel add further control for adding or removing variables from the current tour. The active variable space is the subset of variables currently selected, and their variable circles are drawn with a bold outline. When a variable is deselected on the variable circles the variable fades out gradually, to maintain continuity of motion.

The variable projection coefficients can be manually manipulated using manual controls. To select the variable to manipulate, click on the purple **Manip** button and then click on the variable circle. Once a manipulation mode has been selected, horizontal mouse motions in the plot window alter the coefficient for the manipulation variable, constrained by the values of the coefficients of other active variables (which may also change).

There are 5 manipulation modes: *oblique* allows unconstrained manipulation, *horizontal* and *vertical* constrain manipulation along the axes, *radial* constrains manipulation to the current direction of the variable keeping angle fixed, and *angular* manipulation allows rotating the variable axis in the plane of the plot window, keeping the length of the axis fixed.

The variable axes can be toggled on or off using the options menu at the top of the plot window.

To activate this view mode from the keyboard, type *g* or *G* with the focus in the plot window, or type Control-*g* or Control-*G* with the focus in the console.

### 6.4.1 Projection Pursuit

A guided tour is available when the **Projection Pursuit** button is selected. It is controlled through a separate pop-up projection pursuit window, which contains a plot of the projection pursuit index. When **Optimize** is selected, the tour is guided by the index rather than proceeding randomly. The numbers displayed to the right of the **PP index** label are the minimum, current value, and maximum of the index. A selection of indices is available.

Often, sphering the data ahead of time provides more interesting results with the 2D guided tour, especially for the holes and central mass indices. Use the **Tools** menu and use the **Sphering...** tool to clone sphered counterparts of the currently active variables to do this.

### 6.4.2 Options

The **Options** menu on the main menu bar contains controls for laying out variable circles in the variable selection panel in different ways, and also for toggling variable fading on or off. Variable fading means a variable smoothly fades out when it is de-selected. The alternative is to zero the variable out of view immediately, which creates a discontinuity in the tour motion, but is desirable for some situations.

## 6.5 Rotation: 2D Tour with Three Variables

The rotation mode is essentially a 2D tour that is restricted to use three variables. Its graphical user interface is a subset of the 2D tour interface, with the exception that the three axes are individually represented by toggle buttons labelled **X**, **Y** and **Z**, principally so that it's possible to unambiguously specify the variable to be replaced when selecting a new one.

## 6.6 2x1D Tour

The 1x1D tour generates 2 independent continuous sequences of 1D projections of 2 active variable spaces, plotting the results horizontally and vertically generating a scatterplot. The scrollbar at the top of the controls allows the speed of rotation to be adjusted. The pause checkbox stops and starts the tour. **Reinit** initializes the tour to the projection of the first two active variables. **Scramble** sets the view to a random projection.

Variables can be toggled into the tour by clicking on the variable circles. A click with the left mouse toggles a variable in the horizontal direction, and a click with the middle mouse toggles a variable in the vertical direction. The active variable space is the subset of variables currently selected, and their variables circles are drawn with a bold outline. When a variable is toggled out of the tour it fades out gradually, to maintain continuity of motion.

The variable projection coefficients can be manually manipulated using manual controls. To select the variable to manipulate, click on the purple **Manip** button and then click left or right on the variable circle. Once a manipulation mode has been selected, mouse motions in the plot window alter the coefficients for the manipulation variable or variables, constrained by the values of the coefficients of other active variables (which may also change).

There are 4 manipulation modes: *combined* changes both horizontal and vertical manipulation variable coefficients, *equal combined* constrains the horizontal and vertical changes to be equal, *horizontal* and *vertical* constrain manipulation in the corresponding direction.

The **Options** menu on the main menu bar contains controls for laying out variable circles in the variable selection panel in different ways, and also for toggling variable fading on or off. Variable fading means a variable smoothly fades out when it is de-selected. The alternative is to zero the variable out of view immediately, which creates a discontinuity in the tour motion, but is desirable for some situations.

The variable axes can be toggled on or off using the options menu at the top of the plot window.

To activate this view mode from the keyboard, type *c* or *C* with the focus in the plot window, or type Control-*c* or Control-*C* with the focus in the console.

## 6.7 Scaling of axes

There are three ways to perform view scaling: **Drag** and **Click** scaling are visible in the interface, and available in all viewmodes. When axes are present in scatterplots, you can also scale the view using the axes themselves.

The two styles of interaction, **Drag** and **Click**, are quite different. Drag-style scaling is a perfect example of a direct manipulation interface, in which the points follow cursor motion in a very simple way. However, if you're looking at a lot of data, the points may sometimes lag behind the cursor motion, making the degree of panning or zooming hard to control. There's also no way to hold the aspect ratio fixed with drag-style scaling.

Click-style scaling may take you a few minutes to get used to, but you'll find that it gives you very precise control and is especially useful when you have a lot of data.

To activate this view mode from the keyboard, type *s* or *S* with the focus in the plot window, or type Control-*s* or Control-*S* with the focus in the console.

### 6.7.1 Drag

For the default setting, Drag, the actions of the mouse can be described in terms of a camera: you're operating a camera and looking at a projection of the data in the viewfinder. When you use the left button, the camera is panning freely around, following the mouse exactly. When you use the middle or right button, you're zooming the camera in and out, and changing the aspect ratio of the plot. (There is an option that allows you to fix the aspect ratio, too.)

### 6.7.2 Click

When you select the Click interaction style, the manipulation is not so direct, but your control of the panning and zooming becomes more precise.

With **Pan** selected, the mouse controls the endpoint of a line segment which is anchored at the center of the plot (just where the center of the crosshair is in Drag style). When you press the **Space** bar, you'll cause the plot to pan so that the endpoint becomes the new center – ie, short segments yield small movements. Repeated presses repeat the motion in the same direction – convenient for browsing time-dependent data, for instance.

With **Zoom** selected, the visual guide changes again: this time, the mouse controls a rectangle, and two keys are used: $<$ to zoom in and $>$ to zoom out an amount inversely proportional to the size of the rectangle – that is, large rectangles yield small movements.

### 6.7.3 Reset

To reset the plot, use the menu marked Reset in the main menubar: it has two entries, allowing pan and zoom to be reset separately.

### 6.7.4 Click: pan and zoom options

By the default, the panning and zooming of the plot is unconstrained, moving or rescaling vertically and horizontally with each action. The pan and zoom options allow it to be constrained so that only one axis is affected, convenient for browsing one variable at a time.

### 6.7.5 Using the axes

To pan the data in simple 1D and 2D projections, simply drag the mouse in the axis widgets with the left button pressed. To zoom, drag with the middle or right button pressed.

At this writing, we don't require that ggobi be in the scaling viewmode to scale this way, but we may add that restriction if this raises difficulties for us as ggobi programmers.

## 6.8 Brush: brushing of points and edges

Brushing is often performed when only a single display is visible, but it is most interesting and useful to perform brushing with more than one linked display showing different views of the same data.

To interactively paint points, drag left to move the "brush" within the plotting window, or drag middle to change the size or shape of the brush while you paint. (If you lose the brush by pulling it outside the plotting window, you can grab it again if you press the left or middle button while the cursor is inside the display window.)

To activate this view mode from the keyboard, type *b* or *B* with the focus in the plot window, or type Control-*b* or Control-*B* with the focus in the console.

### 6.8.1 Brush on

When the **Brush on** checkbox is checked, moving the brush over a plotted point causes that point to change its color or plotting character (called a glyph). If the brush is turned off, the brush can be freely moved across the plotting window and it does not change the points.

This is useful if you are plotting a very large number of points, and you want to position the brush before painting, because you can move it much more quickly across the plot.

### 6.8.2 Point and edge brushing

If **Point brushing** is in any state but *Off*, the brush has a rectangular outline. As the brush is moved across the points, any points contained by the brush are affected. You may be changing the color, glyph shape, glyph size, or the "visible" state of the point, depending on the menu setting. The brushing style called "Shadow" deserves a few words: When it is selected, the brushed points are drawn in a color that's very close to the background color [1]. The are de-emphasized but they provide context for the rest of the data.

Similarly, if **Edge brushing** is in any state but *Off*, the brush includes a crosshair, and as the brush is moved in the window, any edges (line segments) intersecting either the vertical or horizontal "hair" are affected. You may be changing the color, line type, line thickness, or visible state of the edge.

If both **Point brushing** and **Edge brushing** are on, the brush is drawn as a crosshair inside a rectangle, and both point and edge brushing are performed.

### 6.8.3 Brushing modes: persistent, transient

There are two brushing modes.

**Persistent**: When you brush a point or edge, it retains its new characteristics when the brush has moved away.
**Transient**: As the brush moves off a point or edge, it returns to the characteristics it had before it was brushed.

### 6.8.4 Undo

Clicking on the **Undo** button restores the characteristics of all points and edges painted between the last mouse-down and mouse-up.

### 6.8.5 Choose color & glyph ...

Clicking on this button opens the **Choose color & glyph** panel, which can be used to choose the point color and glyph as well as the edge type for brushing. At the top of the panel, there is a table of all possible point glyphs and another table of all possible edge types. Clicking on a point glyph sets both the glyph and edge type.

The reason that glyphs and edge types are linked is that points in one display may be linked to edges in another, and then brushing a point with a new glyph may case a linked edge to acquire a new edge type at the same time. (Since there are more point types than edge types for now, it's clear which edge type to select if a new glyph is chosen, but it isn't clear which glyph to select if a new edge type is chosen. For that reason, the edge symbols don't yet respond to button clicks.)

Below those tables is a row of rectangles of color which represent the current color scale. Clicking on one of these sets the brushing color. Double-clicking on one of these rectangles, or on the two rectangles just below them for the background and accent colors, opens a color selection widget with access to the full color map. The **Reverse video** button allows you to swap the background and accent colors.

### 6.8.6 Link by ID or by variable

This option menu is used to define which of two linking rules is to be used. The default rule, *Link by ID*, dictates that points representing records that have the same *id*, as specified in the XML description (or in the API), will respond identically to brushing events. Ids are unique within a dataset, so this rule has no effect when only a single dataset is being studied.

The second rule, *Link by variable*, uses the levels of a categorical variable to link points. To choose a variable, open the **variable manipulation** tool, and select a categorical variable. Now when a case is brushed in one display, all cases with the same value of the categorical variable will change accordingly, in this and all other displays.

For example, look at the *algal-bloom.xml* data supplied with GGobi. It contains four datasets, one of which contains the levels of a factorial experimental, while another represents the response. Both include the same categorical variables. Open two scatterplots, one for each of those two datasets. In the measurements display, plot algal count against day. In the plot of experimental conditions, plot the level of phosphorus against the level of carbon. Prepare to brush in the plot of experimental conditions. Choose brushing by variable, and select "Carbon." Highlighting the low values of carbon, note that all of the points highlighted in the measurements display are among the lowest in algal count.

### 6.8.7 Color schemes ...

Since this button is also found on the Tools menu, it is documented in section 7.5.

### 6.8.8 Color & glyph groups ...

Since this button is also found on the Tools menu, it is documented in section 7.6.

### 6.8.9 Options menu

When the brushing mode is active, the Options menu in the main menu bar includes this item:

**Update brushing continuously**: Update linked brushing with every mouse motion. The alternative is to update linked views only when the mouse is released, which is more efficient when there are a great many points in the plot, or a great many plots on the screen.

### 6.8.10 Reset menu

When the brushing mode is active, the Reset Menu in the main menu bar contains these items:

- Exclude shadowed points in current display: Excluded points aren't drawn, and the views are scaled without them. Excluding a lot of points from large data sets can improve the performance of many operations.
- Include shadowed points in current display: Redraw these points as shadows, and include them in view scaling.
- Un-shadow all points in current display: Restore the points to their usual colors.
- Exclude shadowed edges in current display: As is the case with points, excluding a lot of edges from large data sets can improve performance.
- Include shadowed edges in current display: Redraw these edges as shadows.
- Un-shadow all edges in current display: Restore the edges to their usual colors.

- Reset brush size: Reset the brush to its default size and position.

The first few items in that list may affect more than the current display. Because they really operate on the *data* in the current display, all other displays showing the same data will also respond.

## 6.9 Identification

This mode is used to display labels near points in the plotting window. To see these labels, simply move the cursor inside the plotting window. The label of the point nearest the cursor is displayed. The possible labels are

- the record (case) label supplied by the user either in ASCII or XML (the default),
- the record number (the backup default in case no record label was supplied),
- a list of variables and variable values, where the variables are specified in the list widget above, or
- the record id.

Identification in one window is instantly reflected in all linked windows. [Some thought is required before deciding how or whether this view mode should reflect the linking rules used in brushing.]

To cause a label to become "sticky," click left when the target label is displayed. The printing style changes and the label now remains printed as the cursor moves off, and even remains printed as you leave the **Identify** mode. It is possible to rescale or rotate data, and the sticky labels will continue to be displayed next to their associated points.

To cause a label to become "unsticky," return to the **Identify** mode and click left again when the target point is nearest the cursor. It is also possible to restore all labels to unsticky status by clicking on the **Remove labels** button. You can also see all the labels at once by clicking on **Make all sticky**.

Notice that once a point's label is sticky, you can click **Recenter** to make it the center for the rotation and tour modes.

To activate this view mode from the keyboard, type *i* or *I* with the focus in the plot window, or type Control-*i* or Control-*I* with the focus in the console.

## 6.10 Edit Edges

Here we add points or edges to the datasets by adding them to the displays.

Adding edges: Press the mouse button when the cursor is near the source node, and drag it around the window. You'll see a temporary edge between the source and the nearest node, drawn using the current color and "glyph." When you release the button, one of two things will happen: If you pressed the left mouse button, a dialog window will appear with default values describing the new edge; if you pressed the middle or right button, the edge will simply be added with those same default values.

Adding points: Simply click a mouse button when the cursor is where you want the new point to be located. As above, the left button raises a dialog, and the other buttons simply add the point.

The default values that are assigned are the record label and record id (often the same), and the variable values (if the dataset being augmented has variables). By default, the record label and record id are simply the new record number (represented as a string). If this string already exists as a record id, the new record will not be added.

The default variable values are assigned based on where you clicked on the screen and on the current projection. For any variables not part of the current projection, the default value is 0.

Edge and point deletion have not been implemented. For now at least, you'll have to shadow brush any unwanted elements to get rid of them.

To activate this mode from the keyboard, type *e* or *E* with the focus in the plot window, or type Control-*e* or Control-*E* with the focus in the console.

Note: This mode and the next, "Move Points," may seem like peculiar, even dangerous, additions to data analysis software. They were initially added for the use of another community of xgobi users: discrete mathematicians use xgobi and ggobi to visualize graphs. In that context, moving and editing graphs is quite natural – as it sometimes turns out to be in the context of data analysis, too.

### 6.11 Move Points

In this mode, points or groups of points can be moved. Move the cursor in the window until it's nearest to the point you want to move, then press any mouse button and drag until the point is where you want.

The **Direction of motion** menu allows the movement to be constrained. If the **Move brush group** checkbox is checked, then all points with the same glyph and color as the selected point will be moved with it.

The **Undo last** and **Reset all** buttons allow movement to be reversed.

To activate this view mode from the keyboard, type *m* or *M* with the focus in the plot window, or type Control-*m* or Control-*M* with the focus in the console.

# 7 Tools

## 7.1 Variable manipulation tool

This powerful tool is opened by selecting the first entry on the **Tools** menu. It has several important functions:

- the display of variable statistics,
- variable subset selection for launching multi-plot displays,
- setting variable ranges,
- cloning variables, and
- adding other new variables.

Its first purpose is to report information about each variable. It begins by separating the variables by type: variables are currently classified as categorical or real, though more types can be specified in XML. Categorical variables are displayed hierarchically, and the information reported includes the number of records for each level. For "real" variables, ggobi report the current variable transformation (if any); the minimum, maximum, mean, and median of the raw data; the number of missing values per variable.

Its second purpose is to specify subsets of variables to be plotted when launching a parallel coordinates or scatterplot matrix display. Variables are selected by highlighting rows, and the control and shift keys are modifiers that allow multiple rows to be highlighted.

These selected variables will also respond to operations contained within the panel: you can reset the variable ranges that are used for projecting the data into the plotting window. This allows variables with the same units or potential range (such as percentages) to use the same range, and facilitates visual comparisons.

The selected variables can also be cloned, and the new variables you create will be added to the table as well as the console.

There's another way to add new variables, and that relies on the *New ...* button, which brings up a small panel. Use that panel to specify the variable's name and to set its values: either the row numbers or a set of integers reflecting the assignment of a group identifier to each combination of point color and glyph.

## 7.2 Variable transformation tool

The first step in variable transformation is to specify the variables you want to transform.

There are three stages in the transformation pipeline, with a transformation function in each stage operating on the output of the previous stage. It's equally acceptable to use any or all of them.

You can think of stage 0 as a domain adjustment stage: if a variable has negative values, for instance, many transformation functions can't be applied to it, so you may need to add an increment to each value.

Stage 1 transformations include the Box-Cox family of linear transformations $T(X) = (X^{\lambda} - 1)/\lambda$ [2], and you can either type the Box-Cox parameter into the text box and hit return, or use the spin button to gradually increase or decrease the parameter.

Many of the stage 2 transformations are not linear; they include sorting and ranking.

## 7.3 Sphering

To sphere one or more variables, first select the variables in the list at the top of the window, then click on **Update scree plot**.

It's common to standardize the variables before sphering, that is, use the correlation matrix instead of the variance-covariance matrix. The check box **Use correlation matrix** allows for this option.

Now you're ready to sphere the selected variables. Working your way down the panel, use your visual interpretation of the scree plot together with the information in the labeled section "Prepare to sphere" to decide how many principal components you want to create. By default, all the selected variables will be sphered, but you decide that the first few principal components account for a sufficiently high proportion of the variance. In that case, you can use the spin button to the right of the label "Set number of PCs" to decrease the number of principal components you're going to generate. The variance and condition number are displayed to help you make that choice.

Once you're satisfied with the selected variables and the number of principal components, proceed to the last step. Click on *Apply sphering* to create new variables and add them GGobi's variable selection panels. The names of the selected variables will be added to the "sphered variables" to help you remember which variables you sphered.

## 7.4 Jittering

Select *Variable jittering ...* to open a panel that allows random noise to be added to selected variables. This ameliorates overplotting in scatterplots of data with many ties.

First specify the variables you wish to jitter. Choose between uniform and normal random jitter, and then set the degree of jitter using the slider. To rejitter without changing the degree of jitter, simply click on the **Jitter** button.

## 7.5 Color schemes

The **Color schemes** tool has two purposes: to select a new color scheme, and to automatically color points using the current color scheme.

### 7.5.1 Specifying a color scheme on the command line

If you have specified a colorschemes.xml file in your .ggobirc file (Note: I don't know how to do this under Windows.), then you can also specify a color scheme on the command line when you start ggobi. Use the name of the color scheme, which appears in the XML file, like this:

```
  ggobi -activeColorScheme "ColorScheme Name"
```

or

```
  ggobi --activeColorScheme="ColorSchemeName"
```

Warning: If you launch ggobi with a color scheme which does not have as many colors as you're specifying in the XML data file, strange things will happen; ggobi will attempt to catch this and warn you, but we aren't quite sure what it should do in this case.

### 7.5.2 Choosing a color scheme interactively

Preview a new color scheme by selecting from the tree at the left. If you would like to apply it, click on the button "Apply color scheme to brushing colors." That replaces the current set of colors visible in the "Choose color & glyph" menu and used in all the displays.

If you are currently using $n$ colors and the color scheme you have selected has fewer than $n$ colors, you'll be prohibited from applying the new scheme. If you want to use that scheme, you must use brushing to reduce of colors in use.

The menu of color schemes is populated using a file specified in your .ggobirc file. Here is a minimal .ggobirc file, in which the relevant line specifies the file `colorschemes.xml`, which is included in the GGobi distribution. (You will probably need to modify the path name.)

```
<?xml version="1.0"?>
<!DOCTYPE ggobirc>
<ggobirc>
<preferences>
  <colorschemes file="/usr/ggobi/data/colorschemes.xml" />
</preferences>
</ggobirc>
```

The sample file presently contains 250 or so color schemes, of different types and sizes, and they're based on the work of Brewer ([3], www.personal.psu.edu/cab38). The four types represented are

- diverging: used when the range of the coloring variable has a meaningful midpoint;
- sequential: used to highlight a continuous progression of values;
- qualitative: used when the coloring variable is categorical;
- spectral: Brewer's modifications of the popular spectral scale to reduce its drawbacks. She has made the perceptual steps more uniform, and made the scales more friendly for people with color vision impairments.

### 7.5.3 Applying the color scheme by variable

To brush the data according to the values of a variable, first select a variable in the list at the top of the tool window. This adds two sets of numbers to the display: along the bottom of the display, at the center of each stripe of color, is shown the number of points that will be drawn with this color once **Apply color scheme by variable** is pressed. Along the top of the display, at the boundaries between the stripes of color, appear the values of the chosen variable that define the boundaries between colors.

There are two methods available for defining the bin boundaries, and a menu for choosing between them. The "constant bin width" method simply partitions the range of the selected variable into equal-sized sub-ranges, and maps points into those bins. The "constant bin count" method attempts to map the values into $n$ bins of equal size. Since it also tries to assign all equal values into the same bin, it usually doesn't produce uniform bins if the variable has many equal values.

To adjust the boundaries between stripes of color, grab one of the sliders, and notice that both the values and the counts adjust as you move it. The displays will respond as the sliders are moved: either continuously, or only when you release the mouse. If you your data set has a large number of cases, continuous updating will lag behind the mouse, so it is probably more effective to update only on mouse release.

Try it with the olive oil data used in the tutorial, and select the *Area* variable. It can be a real time-saver.

It's not clear that this tool knows the best way to handle categorical variables yet.

## 7.6 Color & glyph groups

The **Color & glyph groups** tool displays a table. Each unique symbol in the data (combination of color and glyph) occupies a row, and for each row has a toggle button that shadow brushes all points drawn in the corresponding symbol. The three remaining columns report the number of cases shadow brushed, the rest of the cases, and the total number of cases in the group.

Clicking on one of the little symbol displays will assign the currently selected color and/or glyph (depending on the state of the Point Brushing menu on the Brush control panel) to all the points in the corresponding cluster. If your selection would collapse two clusters into one, it will refuse to go ahead: it seems highly likely that someone might do that by mistake, and unlikely that they would do it deliberately. (That choice should be resolved by a dialog, probably, and an 'undo' button should be added.)

The **Exclude shadows** button at the bottom of the window will exclude all shadow-brushed points from consideration in the displays: the views will be scaled without those points, and they won't be drawn. (Excluding a lot of points from large data sets can improve the performance of many operations.) The **Include shadows** button will bring those points back into the plot, and they'll be drawn in the shadow color as before.

The **Update** button updates the contents of the table in case it isn't responding properly to changes in the displays as you continue to brush.

See 7.1 to read how you can add a new variable to the existing data which serves as an indicator for these "clusters."

## 7.7 Subsetting

Select **Case subsetting and sampling ...** to open a panel that allows subsets to be specified in one of six ways.

**Random sample without replacement**: Specify the number of cases to be in the sample.
**Consecutive block**: Specify the first and last row of the block. (The two controls in the second row control the increment used in the control in the first row.)
**Limits**: Use the limits defined by the user in the variable manipulation table to define the subset.
**Every nth case**: Specify the interval and the first row.
**Sticky labels**: All cases with a "sticky" label will be in the subset. If no points have a sticky label, this will have no effect. (See *Identification* for a description of sticky labels.)
**Row labels**: Type in a string, and specify where it should fall (or not fall) in the row labels, and whether case should be ignored. Matches will be included in the subset.

Select one of those six, then click on **Subset** in the bottom row of the panel. If you want to re-include all rows, select the **Include all** button in the bottom row.

Click the **Rescale** button to rescale all plots excluding the points not in the subset.

One purpose of subsetting is to allow the use of GGobi on data matrices that are so large that dynamic and interactive operations begin to become painfully slow. By selecting a smaller subset, a user can work on that subset at a comfortable speed of tour motion and interaction. Another purpose is to do graphical cross-validation: if the feature you see is still there in repeated subsamples, there's a good chance it's not just an artifact of visualization.

## 7.8 Controls for missing data

When your data includes missing values, you can add a new dataset to the current ggobi using the button at the top of the **Missing data** tool. The resulting dataset has the same number of rows as the original, but it may have fewer variables, since it only uses those variables which actually have missing values. It has the same variable names, row labels, glyphs and colors as the original. The difference is this: if the original data in position $i, j$ is missing, the new dataset's value is 1.0; otherwise it's 0.0.

Once the new dataset has been created, plots of missingness information can be launched and linked to plots of the data. The missings plots are pre-jittered to spread the points; the degree of jittering can be later adjusted with the jittering tool.

Linking missings plots to displays of the data allows us to explore the joint distribution of missing values across variables.

### 7.8.1 Imputation

By default, missing values are assigned the value 0, but you may sometimes find that to be an inconvenient choice. Use the **Missing values** panel to assign alternative numbers.

Select the variable or variables whose imputed value you wish to change, then select a method and fill in the text window if necessary, and click **Impute.**

Below the list of variables, a notebook widget allows you to specify the type of imputation you'd like to use.

**Random imputation**: Sample from the present values for each variable to populate the missing values. If you have done some brushing to partition the cases, you can specify that you want the sampling to be done using only cases brushed with the same color and glyph.
**Fixed value**: Specify any value to use instead of the default.
**Percentage below minimum**: Specify a value that is $x$ percent below the minimum value. For example, if the variable ranges from 40 to 80, specifying 10 will assign the missings the value $40 - (10\% * 40) = 36$.
**Percentage above minimum**: Specify a value that is $x$ percent above the maximum.

Click the resale button when you want to update the view to respond to the new values.

For more complex imputation, consider using GGobi from within R.

# 8 Multiple datasets

Several datasets can be open in GGobi simultaneously. The **File** menu interfaces reading in additional data sets. Data tabs corresponding to each open data set appear above the variable selection window on the main controls window, and in various tool windows. The rules for linking these data sets is described in Section 6.8.6.

Several of the sample datasets included with ggobi use multiple datasets. In most cases, the additional datasets are used to add edges, but *algal-bloom.xml* contains four datasets, one of which contains the levels of a factorial experimental, while another represents the response. These two can be linked by variable.

# 9 Edges

A special case of the use of multiple datasets is use of edges (line segments). There are many reasons one would want to display line segments in a scatterplot.

There are several datasets with edges distributed with GGobi: *algal-bloom.xml*, in which edges are used to structure the display of an analysis of variance; *buckyball.xml*, data describing a graph – a geometrical object; *eies.xml*, social network data; *pigs.xml*, a dataset which includes several time variables; *prim7.xml*, in which line segments are used to illustrate structure in the data which was found during extensive exploratory data analysis.

We specify specify line segments (edges) for GGobi by the addition of a second dataset in an XML file (or through the API) in which each record has tags for *source* and *destination.* These edge datasets can still have variables, of course, which might represent variables measured on transactions or interactions.

A scatterplot which has corresponding edge datasets (which we might call edge sets) will have an **Edges** menu in its menubar. If there's a choice of edge sets, you'll see a cascading menu showing their names. Edges can have "arrowheads" added, to indicate the edge's direction; it's also possible to see the arrowheads alone.

Edges can be brushed directly, as described in 6.8.

If the records that define each edge also have variables, then displays of the variables in those edge datasets are also possible. A point in the scatterplot of an edge dataset corresponds to the same record as an edge in another scatterplot. So brushing an edge in one is the virtually the same action as brushing a point in another.

Two plugins, GraphLayout and ggvis, offer methods for laying out graphs. They are documented elsewhere.

## 10 Large data

There exist at least two meanings of "large" in data analysis: large $N$ (number of cases) and large $P$ (number of variables).

### 10.1 Large $N$

We won't attempt to define "large," because it depends so much on your computing hardware and software, but we do know something about the sources of sluggishness, and some effective workarounds.

First of all, there are inefficiencies in the Windows implementation of `gdk`, the drawing library underlying the `gtk+` toolkit in which GGobi is written. Windows users should restrict themselves to rectangular glyphs or point glyphs to get the best possible performance. They certainly should avoid the use of circles. (We have tried to persuade the folks who port `gdk` to Windows to do a better job, but we haven't yet managed to convince them it's important.) Even in dialects of UNIX, single-pixel points are drawn faster than other glyphs.

Use subsets when possible, particularly with the rotation and tour methods. Use the Tools-¿Case subsetting and sampling panel to select a sample of the data.

If the touring and rotation methods become too slow, pause the movement and use the **Scramble** button to see different views.

Avoid the 1-D plotting methods in scatterplots; they both become slow as the data gets very large. The barchart/histogram display is of course much less affected by $N$, and is probably easier to read because it isn't affected by overplotting.

There are a few ways to improve brushing performance, most of which involve postponing linked updates.

- In Brush mode, open the Reset menu in the main menubar and turn off "Update brushing continuously." In that case, linked windows will only be redrawn when you lift your finger off the mouse button. Sometimes you can go even further and perform all your brushing operations with only a single display open, and open the other displays afterwards.
- Turn off the brush until you have it shaped and positioned.
- Use the API instead of the GUI. Using the Rggobi package, for example, you can brush a region in a single command if you can describe it in terms of variable values or indices.

- Working with edges slows down the drawing routines, so you might remove edges during point brushing (or rotation), and then restore them.

# 11 Differences from XGobi

In this section, we summarize the key differences between GGobi and XGobi for those readers who are already familiar with XGobi.

## 11.1 Multiple displays

The first thing you'll notice when you look at a GGobi display is that the plotting window has become separated from the control panels. The main reason for that change is so that a GGobi process can have multiple display windows, of the same type or of different types. In addition to the basic scatterplot, GGobi currently has scatterplot matrices, parallel coordinates plots, and time series plots. (See 3.2 for more detail.)

This design change has had far-reaching effects.

First, user interactions available for the simple scatterplot display can now be made available for other display types. In xgobi, for instance, there is a parallel coordinates display, but it's not possible to brush it – in ggobi, it is.

Unfortunately, having multiple displays introduces a new source of ambiguity: you now have to tell ggobi which display, and which plot within a display, you want to address. Do that by simply clicking inside the target plot. GGobi will draw a thick white outline around that plot so that you can check which plot your actions will be addressing. The ViewMode control panel in the ggobi console should always correspond to the state of the current display, too. We need more user experience before we can tell whether this approach is satisfactory.

The basis for linking has changed, too. All displays of the same data are now linked by default, so it's no longer necessary to run multiple processes in order to achieve linked displays.

One of the most interesting implications of using multiple displays is that a GGobi process is no longer restricted to a single data set. The XML file format makes it easy to specify two or more data matrices in a single file, as described in section 4.2. In addition, it's possible to add data matrices using the Read button on the File menu or using R. (The R - GGobi interface will be introduced below, and it's more fully described in section 5 and in other documentation.)

The rules for linking in XGobi had evolved into a rather complicated hodge-podge with special handling of "row groups," the "nlinkable" notion to exclude points from linking, and linking points to line segments (edges). This has been replaced with a single set of rules that can be specified in the XML file. See section 6.8.6 for details.

With multiple displays, too, we no longer have to launch a new process to open missing value plots – the plots of 1's and 0's which represent the presence and absence of data in each cell.

A convenient side effect of multiple displays is that, since each display now sits in a window of its own, it's now very simple to adjust the aspect ratio of a plot; this simple operation is very awkward in XGobi.

### 11.2 Data format

Before we describe other key changes in the visible design, we'll introduce the changes in data format. The XGobi format, in which a set of files with a common base name is used, is still supported, though some file formats have changed (*.colors*), and some are no longer supported (*.bin*, *.vgroups*, *.lines*). (See section 4 for more detail.) The functions served by the discarded file types are usually served now in a different way: for example, XGobi uses the *.vgroups* file to force a group of variables to use the same axis ranges. In GGobi, that's accomplished by setting limits in the variable manipulation tool.

The new format, in which all the data lives in one file, is written in XML. XML (Extensible Markup Language) is a widely used language for specifying structured documents and data to be viewed and exchanged. It was initially intended to be read by browsers, but it is also used to define documents that are read by other software. XML files can be validated automatically, and XML specifications can be easily extended, too, by adding to the set of tags in use.

We strongly favor the XML format, and we won't try to keep the old format up to date. For instance, it's no longer possible to specify edges in the ascii data format, but they can be specified in XML – and they can have associated data values.

Several XML data files are included in the sample ggobi data, and the details of their format is described in *XML Input Format* (XML.pdf) which is included as part of the ggobi distribution.

Another innovation in data access is the ability to read data directly from a MySQL database, as described in section 4.3.

### 11.3 Integration

Many xgobi users are also users of the SPlus or R statistics software, and have used the S function which launches an xgobi process viewing S data. Once that launch occurred, the resulting process was utterly independent of its parent. The XGobi authors did some experiments in the early 90s to achieve a more intimate connection, but made little headway with S, though interprocess communication was used successfully with ArcView [15, 16].

With ggobi, that problem has been solved, and it's now possible to have real-time integration between ggobi and a variety of other software environments. An example of this integration is the embedding of ggobi into R (or S), with the addition of a set of R (or S) functions that manipulate GGobi data and displays, resetting data values, projection, and the appearance of the plot.

For more details, see section 5, or read *Using the R-GGobi Link* (in RGGobi.pdf), another piece of documentation included in the ggobi distribution.

### 11.4 Variable selection

There have been a few changes in variable selection. The familiar variable circles are still used for high-dimensional view modes, but we've switched to a simple checkbox interface for plots where only one or two variables can be selected simultaneously. In these plots, the rich feedback provided by the variable circles is not needed, and may just be confusing to novices.

The basics of the user interface haven't changed, though: click with the left button to select a variable to be plotted horizontally and the middle (or right) button to plot vertically.

### 11.5 The variable manipulation tool

The **Variable manipulation** tool presents summary statistics for each variable. If a variable has been described in the XML data file as categorical, it also shows the name and count for each level. The tables can also be used to select variables – not for plotting, but for specifying variables subsets when launching a parallel coordinates or scatterplot matrix display.

This tool can also be used to specify scaling ranges for variables or groups of variables, and so we have dispensed with the *.vgroups* functionality in XGobi.

Variable cloning is another new feature: it appears as a button on the variable selection and statistics table.

For more detail on this table, see section 7.1.

### 11.6 Changes in ViewModes and Tools

**Brush** may be the view mode that has changed the most, because GGobi has a much richer notion of color selection than XGobi. Open the **Choose color & glyph** panel, and then double-click on any element of the color palette to bring up a color selection widget with access to the full color map. Notice that you can change the background color as well as any of the foreground colors, and notice that changing the glyph also changes the line type to be used in edge brushing.

The **Color schemes** tool has extended brushing considerably, by allowing a choice of color schemes and enabling a form of automatic brushing. See 7.5.

**Scale** has also changed. The direct manipulation shifting and scaling methods work as they did in XGobi, but we've added what we call "Click-style interaction" for more precise control. See section 6.7.

The tour methods in GGobi are still evolving: many things are not yet implemented, but new things are beginning to appear. In the **1DTour**, a projection of several variables is viewed as an average shifted histogram, as described in section 6.3.

The redesign of the tour methods is reflected in the **Sphering** tool, which also appears in the most recent versions of XGobi. Instead of automatically sphering variables in projection pursuit, GGobi requires you to specify which variables to sphere. Since this method makes use of variable cloning, it creates new variables, allowing you to look at plots of principal components against the original data.

See section 7.3 for details.

### 11.7 On-line help

The on-line help system used in XGobi has been replaced with "tooltips," so leaving the mouse over a widget for a couple of seconds brings up a phrase describing the function of that widget. If the tooltips annoy you, you can turn then off using a checkbox on the **Options** menu.

## 12 Known problems

## 13 Future work

- Edge editing
- Maps – or the linking of GGobi to existing software for spatial data analysis and display

## Web Links

The web site for GGobi:

`ggobi`: `www.ggobi.org`

contains details on downloading and installing software, related documentation and a picture gallery.

# References

[1] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.

[2] George E P Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society*, B-26:211–243, 1964.

[3] Cynthia A. Brewer. Color use guidelines for data representation. In *Proceedings of the Section on Statistical Graphics*, pages 55–60, Baltimore, 1999. American Statistical Association.

[4] A. Buja, D. Asimov, C. Hurley, and J. A. McDonald. Elements of a Viewing Pipeline for Data Analysis. In W. S. Cleveland and M. E. McGill, editors, *Dynamic Graphics for Statistics*, pages 277–308. Wadsworth, Monterey, CA, 1988.

[5] A. Buja, D. Cook, D. Asimov, and C. Hurley. Dynamic Projections in High-Dimensional Visualization: Theory and Computational Methods. Technical report, AT&T Labs, Florham Park, NJ, 1997.

[6] A. Buja, D. Cook, and D. Swayne. Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996. See also `www.research.att.com/∼andreas/xgobi/heidel/`.

[7] D. Cook and A. Buja. Manual Controls For High-Dimensional Data Projections. *Journal of Computational and Graphical Statistics*, 6(4):464–480, 1997. Also see `www.public.iastate.edu/∼dicook/research/papers/manip.html`.

[8] D. Cook, A. Buja, and J. Cabrera. Projection Pursuit Indexes Based on Orthonormal Function Expansions. *Journal of Computational and Graphical Statistics*, 2(3):225–250, 1993.

[9] D. Cook, A. Buja, J. Cabrera, and C. Hurley. Grand Tour and Projection Pursuit. *Journal of Computational and Graphical Statistics*, 4(3):155–172, 1995.

[10] M. Forina, C. Armanino, S. Lanteri, and E. Tiscornia. Classification of olive oils from their fatty acid composition. In H. Martens and H. Russwurm Jr., editors, *Food Research and Data Analysis*, pages 189–214. Applied Science Publishers, London, 1983.

[11] A. Inselberg. The Plane with Parallel Coordinates. *The Visual Computer*, 1:69–91, 1985.

[12] David W Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley, New York, 1992.

[13] D. Swayne and A. Buja. Missing Data in Interactive High-Dimensional Data Visualization. *Computational Statistics*, 13(1):15–26, 1998.

[14] D. F. Swayne, D. Cook, and A. Buja. XGobi: Interactive Dynamic Graphics in the X Window System. *Journal of Computational and Graphical Statistics*, 7(1):113–130, 1998. See also `www.research.att.com/areas/stat/xgobi/`.

[15] Deborah F. Swayne, Andreas Buja, and Nancy Hubbell. XGobi meets S: Integrating software for data analysis. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 430–434, Fairfax Station, VA, 1991. Interface Foundation of North America, Inc.

[16] J. Symanzik, D. Cook, N. Lewin-Koh, J. J. Majure, and I. Megretskaia. Linking ArcView 3.0 and XGobi: Insight Behind the Front End. *Journal of Computational and Graphical Statistics*, 9(3):470–490, 1999.

[17] John Tukey and Paul Tukey. Strips displaying empirical distributions: I. textured dot strips. Bellcore Technical Memorandum, 1990.

[18] E. Wegman. Hyperdimensional Data Analysis Using Parallel Coordinates. *Journal of American Statistics Association*, 85:664–675, 1990.