

GGobi

Deborah F. Swayne	Duncan Temple Lang
AT&T Labs – Research	Bell Laboratories
Florham Park, NJ 07932	Lucent Technologies
dfs@research.att.com	Murray Hill, NJ

Andreas Buja	Dianne Cook
AT&T Labs – Research	Iowa State University
Florham Park, NJ	Ames, Iowa

GGobi is a new interactive and dynamic software system for data visualization, the result of a significant redesign of the older XGobi system [Swayne et al., 1998], whose development spanned roughly the past decade. GGobi differs from XGobi in many ways, and it is those differences that explain best why we have undertaken this redesign.

GGobi’s appearance: GGobi’s appearance has changed from that of XGobi in several ways: 1) It uses a different graphical toolkit with a more contemporary look and feel and a larger set of components. The new toolkit is called **GTK+**, which explains the initial G in GGobi. 2) With XGobi, there is in general a single plot per process; to look at multiple views of the same data, one launches multiple processes. A single GGobi session can support multiple plots (which may represent multiple datasets), and a single process can support multiple independent GGobi sessions. 3) XGobi’s display types are essentially a single scatterplot and a subordinate parallel coordinate plot, but GGobi supports several types of plots as first class citizens: scatterplots, parallel coordinate plots, scatterplot matrices, and time series plots. A GGobi session in progress can be seen in Figure 1.

Other changes in GGobi’s appearance and repertoire of tools include an interactive color lookup table manager, the ability to add variables on the fly, a new interface for view scaling (panning and zooming), and a redesign of the tour code.

Portability: A major advantage of using the new toolkit (**GTK+**) is portability. It originates in the Linux community, but it has been ported to Microsoft Windows and is being ported to the Macintosh. It is complicated to run XGobi on a machine running Windows, because it requires the installation an X Window System server. GGobi, on the other hand, runs directly under Windows.

GGobi’s data format: GGobi’s data format has been extended significantly from that of XGobi. To describe a set of data for the older XGobi, one creates a set of files with a common base name, with the data in one file, and other files for the labels, colors, glyphs, and so on. GGobi continues to support this scheme in a limited way, but its new format uses a single file in XML, the Extensible Markup Language, which is emerging as a standard language for specifying structured documents and data formats.

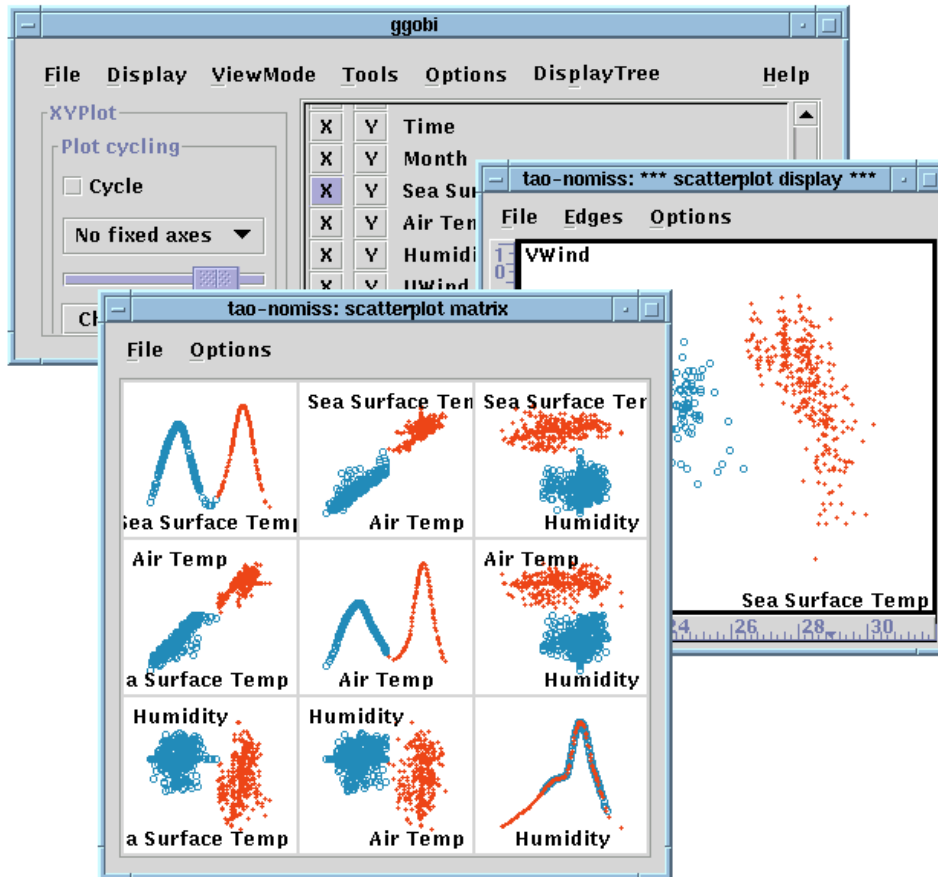


Figure 1: GGobi, a descendant of XGobi, showing a scatterplot and linked scatterplot matrix. GGobi has many differences from XGobi, both visible (multiple displays are possible, and the toolkit has a more modern look) and invisible (it has been designed to facilitate software integration).

The use of a single file aids consistency of the different elements of the input, making it easier to validate and maintain. An XML document looks a bit similar to an HTML document, but it allows one to introduce new markup elements. The use of XML in GGobi allows complex characteristics and relationships in data to be specified. For example, multiple datasets can be entered in a single XML file, and specifications can be included for linking them. Using the XML format, GGobi can read compressed files and can read files over the network.

Interoperability: While GGobi is a stand-alone application, it has been designed and constructed as a programming library so that direct manipulation, dynamic visualization functionality can be embedded within other applications. It has an Application Programming Interface (API) which developers can use to integrate the GGobi functionality with other code. At the highest level, there are three different approaches to integrating GGobi with other software, each of which is described in more detail in [Symanzik et al., 2002]. These approaches create an environment that encourages others to experiment with innovations in data visualization without

the need to build an entire system from scratch.

Embedding GGobi within other Applications: In this approach, we treat GGobi as a programming library and allow its data structures to be compiled into other software to create customized applications. When GGobi is embedded in this way, it can be controlled using programming interfaces from other languages such as Java, Perl, Python and S.

Extending GGobi: The use of modular plugins allows one to dynamically load code into a running GGobi. Using this approach, programmers can add functionality to GGobi without having to dig deeply into the code, and they can share their extensions easily with others. Various authors have been experimenting with plugins to communicate with a database, perform graph layout, and compute a minimum spanning tree, among others.

Distributed/Remote Access: The client/server architecture allows one to create GGobi as a server process offering a variety of methods to control, query and modify the session. Other applications can invoke these methods, even across different machines.

Getting GGobi: The GGobi code, sample data sets, and documentation can be found on www.ggobi.org.

References

- [Swayne et al., 1998] Swayne, D. F., Cook, D., and Buja, A. (1998). XGobi: Interactive dynamic data visualization in the X Window System. *Journal of Computational and Graphical Statistics*, 7(1):113–130.
- [Symanzik et al., 2002] Symanzik, J., Swayne, D. F., Temple Lang, D., and Cook, D. (2002). Software integration for multivariate exploratory spatial data analysis. In *New Tools for Spatial Data Analysis: Proceedings of the Specialist Meeting*. Center for Spatially Integrated Social Science.