



GEOGRAPHIC STATISTICS VISUALIZATION: WEB-BASED LINKED MICROMAP PLOTS

By Xusheng Wang, Jim X. Chen, Daniel B. Carr, B. Sue Bell, and Linda Williams Pickle

LINKED MICROMAP (LM) PLOTS OFFER A NEW TEMPLATE FOR DISPLAYING SPATIALLY INDEXED STATISTICAL SUMMARIES.¹ THIS TEMPLATE HAS FOUR KEY FEATURES: IT CAN

- Display at least three parallel sequences of panels (micromap, label, and statistical summary) that are linked by position
- Sort the study units
- Partition the study units into panels to focus attention on a few units at a time
- Link the highlighted study units across corresponding panels of the sequences

You can use LM plots to visualize complex data in many areas.¹⁻³ This column extends the existing work by introducing *Web-based interactive* LM plots, a statistical data visualization system that integrates geographical data manipulation, visualization, interactive statistical graphics, and Web-based Java technologies. The system effectively presents the complex and large-volume sample data of national cancer statistics of the United States.

Structure and features

Displaying LM plots on the Internet introduces a new and effective way to visualize statistical summaries. Through a Web browser, a user can easily access and view the statistical data in LM plots—globally. Effective Web-based LM plots include many interactions that require retrieving new data from the Web server and displaying it in different

layouts to help users. The Java programming environment provides mechanisms that let users interactively control a Web browser's display content, so we used Java as the programming tool in our implementations.

Although spatial resolution is lost in the transition from the printed page to a computer monitor, the interactive viewing options allowed better visualization through drill-down views, multiple levels of detail, sorting, magnified micromaps, miniature overall statistical summaries, confidence interval switching, and other interactive visualization methods. These methods are not new individually, but their integration with LM plots provides a new approach to communicating spatially indexed statistical summaries over the Internet.

We used cancer statistics from the National Cancer Institute. The displays are of test data, not of official cancer statistics, but nonetheless provide an excellent testbed for studying statistical visualization methodology. We implemented a full-fledged set of LM plots for present US cancer statistical summaries at the state and county levels. We restricted the testbed list of selectable cancer sites to cancers of the cervix and of the lung, but can readily extend it. These Web-based interactive LM plots preserve all key features of

the LM plots originally published.

To help you understand our work, as we introduce the structure and features of the Web-based LM plots, we suggest that you access a full-featured interactive demo system on our Web site (<http://graphics.gmu.edu/~xwang/cancer4>). We developed the demo using Internet Explorer and recommend that you view it with that browser. Netscape and possibly other browsers can introduce minor differences.

Display panels and study units

The LM plots for displaying cancer summary statistics have four parallel sequences of display panels (columns): US and State micromaps, State and County names, and two cancer statistical summaries.

The study units (rows) are States or Counties that you can change, depending on the current display content, from a pull-down menu. For national cancer statistics, the study units are States. For state cancer statistics, the study units are Counties. You can view approximately 41 study units in one frame at a time. When the number of study units exceeds this maximum, the display panel becomes scrollable. Figure 1 shows the basic layout of LM plots for US cancer statistics.

In a linked study unit's row, the geographic location, the dot before the name, and the statistics are all the same color. Each statistics panel shows the corresponding statistical value as a dot and its *confidence interval* bounds with line segments on both sides. The default scaling is set to include the lowest

lower bound and largest upper bound of the CIs. However, when a cancer is rare and the population is small, the CIs' upper bound can be extremely large. The CI button, located at the top of each statistical display panel, is an icon depicting a dot with line segments on each side. You can use the CI button to switch between the default scaling that shows a complete CI view and a view that is scaled to the values of the maximum and minimum point estimates truncating the long CIs.

Sorting

Study unit sorting is an important feature of LM plots. When sorting on a statistical variable, the displayed dot curve can clearly show the relative relationships among the study units. By clicking on a button, you can interactively sort the name and statistics panels for fast visualization from several interchangeable perspectives. (A sorting button with a triangle icon is next to the panel's title.) An up-triangle icon on the button represents an ascending sorting, and a down-triangle icon shows a descending sorting.

For example, Figure 1 shows the study units (States) sorted by one statistical summary (cancer mortality rate) in descending order.

Linking

Linking all of a study unit's elements can help to highlight it. One color represents all the elements across the corresponding panels of the sequences. Moreover, if the user moves the mouse cursor over any one of the related elements (micromap, unit name, or colored dots), all linked elements in the study unit will blink. Meanwhile, the corresponding statistical summaries appear in text at the bottom of the browser window (in the browser's status line).

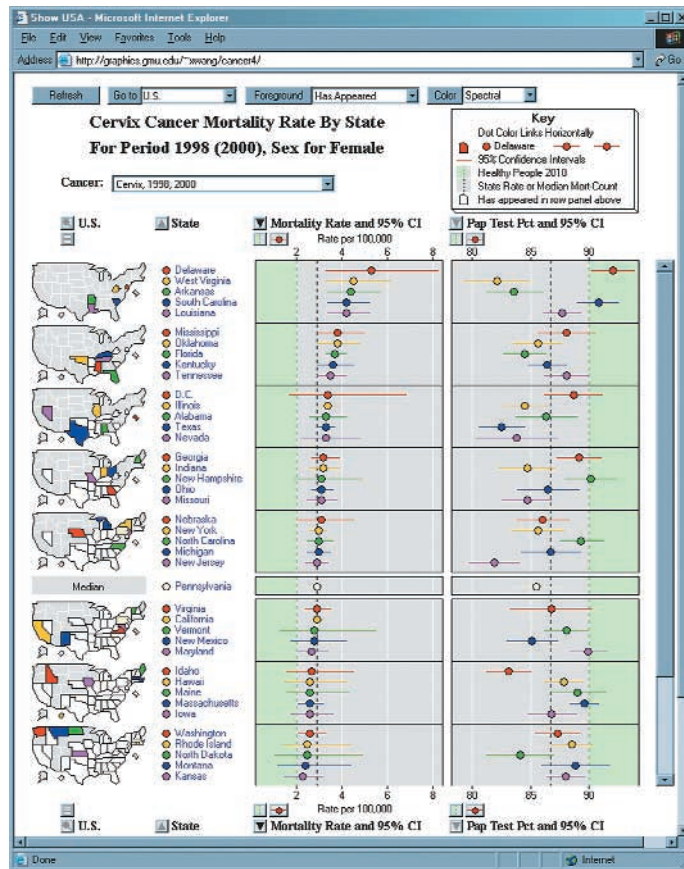


Figure 1. A snapshot of linked micromap plots of US cancer statistics.

Grouping

The study units are partitioned into subpanels vertically to help focus attention on a few units at a time. In our design, we group every five consecutive study units into a subpanel. In each group, each study unit is assigned one of five different colors. This also helps show the linked elements within a group more clearly.

Color

The coloring scheme for micromaps serves multiple objectives.⁴ Simply, it facilitates rapid location of a particular study unit in a subpanel. In our LM plots, we repeat the coloring scheme for all groups. Within a group, the coloring scheme can reflect sequence. Our current design includes three coloring schemes: spectral, sequential, and divergent. The default spectral coloring scheme will work well for most users. The sequential scheme works well for the color blind and for black-and-white

printing. The divergent scheme also works well for the color blind, but not for black-and-white printing. A pull-down menu lets a user interactively select a coloring scheme.

Micromaps and magnification

Micromaps are the most active components in LM plots. Through micromaps, a user can intuitively find the study units' locations. Furthermore, a user can see the geographical distributions of each group's related study units because the grouped study units are highlighted by color in a corresponding micromap.

In a micromap, we divide all study units—except for the colored ones—into two categories: background and foreground. The micromap displays the background study units in a light gray color with a white outline. The foreground study units are displayed in white or light yellow, according to the selected coloring scheme, and outlined

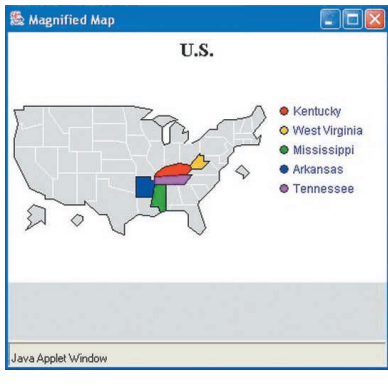


Figure 2. A magnified micromap window.

in black. The black outline overwrites shared white boundaries. Foreground coloring has two uses. First, it provides an index for the user searching the micromaps for a particular area of interest based on geometry. If the area is in the foreground color, it's in a micromap group higher in the sort order than the micromap you're viewing. If the area is in the background color, it's in a micromap group lower in the sort order. Second, the foreground color provides a cumulative geographic view of the ordered data that could show interesting geographic patterns, such as a clustering of higher rates.

A pull-down menu lets users identify which study units belong in the foreground (or background). In the current design, the menu has these choices:

- *Has appeared*: Based on the sort order, the foreground study units in white or yellow have appeared higher in the sort order. The background study units in gray are lower in the sort order.
- *Will appear*: This reverses the “Has appeared” categories, so the foreground study units in yellow or white are lower in the sort order. Similarly, the background study units in gray are higher in the sort order.
- *Above/below median*: For the micromaps above the median unit, foreground study units in yellow or white are above the median unit, and the background study units in gray are below the median unit. For the micromaps below the median unit,

the meanings of the foreground and background study units reverse.

Some states might include many counties (study units). For example, Texas has over 200 counties. Because each micromap occupies a small fixed area, a study unit's area in a micromap might be too small to see clearly. To open a magnified micromap window, you can click the magnification button next to the micromap panel title or right-click a study unit name or the dots in the corresponding statistical summary panels. You can adjust the magnified micromap's size by dragging the window's corner. Moving the mouse over the study unit in the micromap causes the study unit to appear in color in the magnified view. Figure 2 shows a magnified US micromap.

Automatic scrolling

Because the number of study units in an LM plot can exceed the maximum that the monitor screen can hold, all study units might not be displayed in the panel. So, when a user highlights a study unit from a micromap or a magnified micromap and the study unit is not in the display panel, the user will not see the corresponding elements of the unit blinking. In our implementation, when this happens, the user can click the right mouse button, and the display panel will automatically scroll so that the corresponding study unit appears in the display panel. This operation can also help the user rapidly locate a study unit.

Drill-down and navigation

Drill-down lets a user zoom in from a high-level LM plot to the corresponding low-level LM plot to view detailed statistical data and the relationships among different hierarchical levels of detail. When using LM plots

to visualize the national cancer statistical summaries, the US LM plot provides a starting place for subsequent drill-downs to any other state LM plot. In the US LM plot, the state names and all micromap regions are active links to corresponding state LM plots. In general, when moving the mouse cursor over an element, if the mouse cursor becomes a hand, the element is an active link. Clicking on that element drills down to the next LM plot level.

Drill-down is also available from a magnified micromap. When one area in the magnified micromap is blinking and the mouse cursor becomes a hand, clicking the left mouse button will also carry out the drill-down operation.

Our system also allows direct navigation through a pull-down menu from one LM plot to any other LM plot regardless of the current LM plot level. From the menu, a user can directly access the US LM plot or any other state LM plot.

Overall view

Because the user can't always see the whole statistical curve formed by the dots shown in the statistical panels, we added a pop-up window that displays the scaled-down pattern (see Figure 3). Clicking the button below the magnification button activates this display window.

Displaying different statistical data sets

Interactively displaying different statistical data lets users view and compare the relationships among different results. In our system, we provide a pull-down menu for a user to choose and view different statistical cancer data sets. Once a user selects a new cancer type, the data will be downloaded from the Web server and presented in the display panel with a new calculated scale. To

improve display efficiency, the downloaded cancer data will be buffered in memory so that the next time a user selects this cancer type, it will fetch the data directly from memory.

Technical problems

A Web programming environment is stateless, which means that states in the current Web page cannot be brought to the next Web page. For Web-based LM plots, each LM plot is a Web page and so navigating among LM plots means accessing different Web pages. However, users won't want to customize settings for each new page. Therefore, Web-based LM plots must preserve current setup states, just like a stand-alone application that keeps all states for the session's duration. So, once a user accesses an LM plot, all states selected for that LM plot should be preserved to serve the whole session no matter where the user navigates.

Additionally, efficient data retrieval is very important for a statistical visualization application. You can save such data in different formats, places, or modes to expedite retrieval.

In this section, we discuss how our innovative solutions address these technical problems and increase efficiency and use of the Web-based interactive LM plots.

Stateless programming environment

Generally, two methods can solve this problem. The first saves the states in the request sent to the Web server. The second directly saves the states at the Web server through a Web-programming tool like Microsoft ASP. However, both methods require writing special code on the Web-server side. This increases programming complexity and also affects Web-access efficiency. Moreover, this might limit

the Web server platform's independence from Web page content. For example, the Microsoft Web server supports ASP, but the Web-based LM plots developed in this environment might not work for other Web servers.

We can exploit Java's programming capabilities to solve this problem in a new way. First, let's see why a Web programming environment is stateless. The main reason is that the Web browser always uses HTTP to request HTML messages from the Web server. Unfortunately, HTTP itself is a stateless protocol, and HTML is a stateless language. A Web browser always loses all information in the previous Web page when it displays a requested message (HTML file) from the Web server. Under such conditions, when a Web page (actually the embedded code) cannot write any states to the Web client's devices, the program code embedded in one Web page cannot directly pass its states to the code in another Web page, unless there is special code that works on the Web server.

From this, we can see that the solution must be in the program code embedded within one Web page instead of separate Web pages. If the program code embedded in a Web page can directly request messages from the Web server and control the message display in the browser's window, the program states saved in memory by the code will be available as long as the program session is alive. You can retain the states in the Web browser because the program code receives the returned messages instead of the Web browser.

We can design a Java applet to implement this mechanism. Once a Web browser displays a page, the embedded Java applet displays the LM plots and accepts any user input actions located in the applet panel. If the applet detects that one action needs to retrieve new

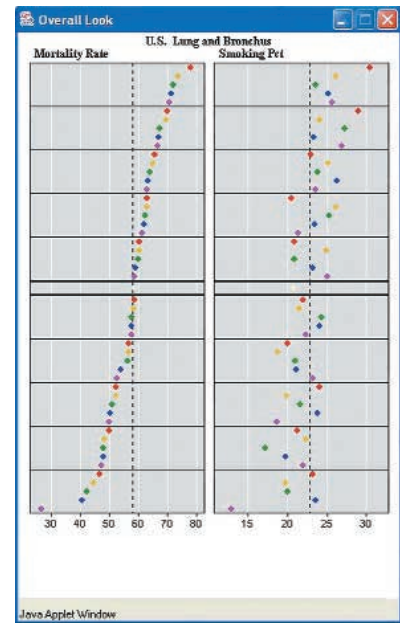


Figure 3. An overall view of the statistical panels.

data from the Web server, it activates the communication code to request certain data from the Web server and displays the data in the applet panel. The result looks like a new Web page, but it's really just a programmed display refreshment. The newly displayed pages are not necessarily HTML pages. Instead, they can be displays on the same Java applet panel, painted by the same Java applet code according to the retrieved data. In application, this process is transparent to the Web users, and users see a new "Web page" displayed after clicking on an active link. (Readers might notice the discrepancy when using "Back" or "Reload" in a Web browser, which invokes a different Web page.)

Statistical data retrieval

The data format (such as binary or text) is the organization of the data flow received by the Java applet. The data format is generally not a problem for Java code, which can handle any data format as long as it has certain rules to follow.

Statistical data can be saved anywhere. It might be on the Web server's system or on a different remote system.

Xusheng Wang is a PhD candidate in the School of Information Technology and Engineering at George Mason University. His research interests include virtual reality, real-time simulation, and low-level graphics algorithms. He is a member of the GMU Computer Graphics Group at George Mason University. Contact him at George Mason Univ., Fairfax, VA 22030-4444; xwang1@gmu.edu.

Jim X. Chen is an associate professor in the Department of Computer Science at George Mason University. He is the Director of the Graphics Lab at GMU and a program cochair of IEEE Virtual Reality 2003. His research interests include graphics, visualization, virtual reality, networking, and simulation. He received a PhD in computer science from the University of Central Florida. He is a member of the IEEE Computer Society. Contact him at George Mason Univ., Fairfax, VA 22030-4444; jchen@cs.gmu.edu; www.cs.gmu.edu/~jchen.

Daniel B. Carr is a professor in the Department of Applied and Engineering Statistics at George Mason University. His research addresses many of the challenges in statistical graphics design, including representation of large data sets, visual data mining, and web communication. He received a PhD in statistics from the University of Wisconsin, Madison. He is a Fellow of the American Statistical Association and a columnist for the joint newsletter of the Statistical Computing and Statistical Graphics sections. Contact him at George Mason Univ., Fairfax, VA 22030-4444; dcarr@gmu.edu; www.galaxy.gmu.edu/~dcarr.

B. Sue Bell is a mathematical statistician in the Statistical Research and Applications Branch, Division of Cancer Control and Population Sciences, National Cancer Institute. She leads a team to design and implement a Web site that will present cancer surveillance statistics using the best available graphics and statistical analysis. Her research interests include spatial statistics and data visualization. She received a PhD from the University of Texas School of Public Health. Contact her at the National Cancer Institute, Bethesda, MD 20892-8317; bellsu@mail.nih.gov.

Linda Williams Pickle is senior mathematical statistician in the Statistical Research and Applications Branch, Division of Cancer Control and Population Sciences, National Cancer Institute. She received a PhD in biostatistics from Johns Hopkins University. She has published several mortality atlases and conducts research in the areas of spatial statistical modeling, geographic information systems, and data visualization. Contact her at the National Cancer Institute, Bethesda, MD 20892-8317.

Regardless of the system's location, as long as it's connected to the Internet, Java code can always retrieve the data through TCP/IP.


We must address the statistical data-saving mode carefully. Currently, two general data-saving modes exist: file or database. If you save data in the file mode, and the read access right is open, the Java code can directly read the data file through the Internet. However, if you save the data in a database system, the situation becomes complex. Generally speaking, through JDBC (Java Database Connectivity), Java code can retrieve the data saved in a database. However, for security reasons, a safety database system is usually not open to the public. Therefore, the Java applet embedded in a Web page

generally can't access a database system through JDBC directly.

We believe the way to access the data is through the Web server, to which the database system might release the access rights. Under such a situation, if a Java applet wants to retrieve the data from the database, it must get aid from the Web server. This means that you must write a data-retrieval code with the JDBC API on the Web-server side. In this way, when a Java applet needs to retrieve data from a database, it first tells the data-retrieval code on the Web server what data is needed. Then, the Web-server code connects to the database system, retrieves the data through the JDBC API, and sends the retrieved data back to the applet code on the Web-client side. Finally, the applet for-

gets the received data and displays the plots on the applet panel. To provide high efficiency, you can code the data-retrieval code on the Web server in Java Servlets.

In our application, we save the statistical data in the file mode and on the Web server. We don't need to write any code for the Web server. Instead, the Java applet directly retrieves the cancer data from the files.

Because the Internet is a widely accessible source of public information, the Web-based implementation of LM plots will make information available to more users, and the new interactivity can lead to more involvement and better understanding. Additionally, with some modifications, you could easily use this method to visualize other spatially indexed statistical datasets over the Internet. 

Acknowledgments

The National Cancer Institute sponsored this project. However, all opinions are solely the authors'. NSF Research No. 9983461 supported portions of the research.

References

1. D.B. Carr et al., "Linked Micromap Plots: Named and Described," *Statistical Computing & Statistical Graphics Newsletter*, vol. 9, no. 1, Summer 1998, pp. 24–31.
2. D.B. Carr et al., "Using Linked Micromap Plots to Characterize Omernik Ecoregions," *Data Mining and Knowledge Discovery*, vol. 4, 2000, pp. 43–67.
3. D.B. Carr, J.F. Wallin, and D.A. Carr, "Two New Templates for Epidemiology Applications: Linked Micromap Plots and Conditioned Choropleth Maps," *Statistics in Medicine*, vol. 19, nos. 17–18, Sept. 2000, pp. 2521–2538.
4. D.B. Carr, "Designing Linked Micromap Plots for States with Many Counties," *Statistics in Medicine*, vol. 20, nos. 9–10, May 2001, pp. 1331–1339.