BIOINFORMATICS APPLICATIONS NOTE Vol. 20 no. 17 2004, pages 3246–3248
doi:10.1093/bioinformatics/bth349

Java Treeview—extensible visualization of microarray data

Alok J. Saldanha

Department of Genetics, Stanford University School of Medicine, Stanford, CA 94305, USA

Received on May 20, 2004; revised and accepted on May 25, 2004
Advance Access publication June 4, 2004

ABSTRACT

Summary: Open source software encourages innovation by allowing users to extend the functionality of existing applications. Treeview is a popular application for the visualization of microarray data, but is closed-source and platform-specific, which limits both its current utility and suitability as a platform for further development. Java Treeview is an open-source, cross-platform rewrite that handles very large datasets well, and supports extensions to the file format that allow the results of additional analysis to be visualized and compared. The combination of a general file format and open source makes Java Treeview an attractive choice for solving a class of visualization problems. An applet version is also available that can be used on any website with no special server-side setup.

Availability: http://jtreeview.sourceforge.net under GPL.

Contact: alokito@users.sourceforge.net

INTRODUCTION

Java Treeview is an enhanced, open source, cross-platform rewrite of the original, windows-only Treeview (Eisen *et al*., 1998). The original Treeview provides a simple interface for viewing the results of hierarchical clustering. The clustering is done by a separate program that creates a tab-delimited text file called a 'clustered data file' or CDT file. The simple structure of the CDT file allows developers to use it as an output format for other tools. Although it was originally developed for gene expression data, Treeview has since been used to view the results of hierarchical clustering of other types of data, including GFP reporter levels and motif significance scores. In addition to making the functionality of Treeview available to a large audience, Java Treeview supports a generalized CDT format that allow many additional details, such as colors of genes, arrays, nodes and heights of terminal branches, to be specified. Furthermore, the generic structure of the generalized CDT and open source nature of Java Treeview encourage not only further enrichment of the dendrogram representation, but also the development of completely novel visualizations, several of which are presented here. These features dramatically expand the utility of Treeview, and have met with an encouraging response; in the 6 months following the 1.0 release, Java Treeview has been downloaded over 4000 times through word of mouth, the majority of times by Windows users with access to the original Treeview.

Comparison of multiple analysis methods is an important task that can be aided by superior visualization tools. New methods of microarray analysis appear almost daily in the literature. Also, to the researcher in the field it is tempting to consider minor modifications to existing methods. Many of these analyses associate a score or annotation with genes, arrays or nodes. The generalized CDT file, described in the manual, provides a natural place to put this information. Java Treeview loads the generalized CDT file into a standard data structure and makes it available for representation within an existing or novel visualization. For example, the location of genes on a microarray may be relevant to an investigation of spatial bias, the location of genes on the chromosome may be of interest in an array CGH experiment, and in many cases arrays may be annotated to distinct classes, for instance in a cancer study different types of source tumors. These additional types of data can be incorporated into the CDT file and represented either alongside the dendrogram or in a separate tabbed display (see Features and Fig. 1). All displays are linked together with a shared selection model, so that genes selected according to a particular criterion in one display are also selected in the other displays. This aids in comparison of the different visualizations, and adds value over special purpose visualization tools.

There are a wide variety of tools available to the genomics researcher. A few examples, include AVA (Zhou and Liu, 2003), TRANSPATH (Krull *et al*., 2003), Genesis (Sturn *et al*., 2002) and J-Express (Dysvik and Jonassen, 2001). However, many of these tools are monolithic applications that proscribe the types of data and analysis that may be done, are available for only a few platforms, or come with commercial entanglements. Java Treeview is thus closest in spirit to the TableView application (Johnson *et al*., 2003), but is built to display large phylogeny-based datasets with hierarchical trees; under Mac OSX, displaying a clustergram

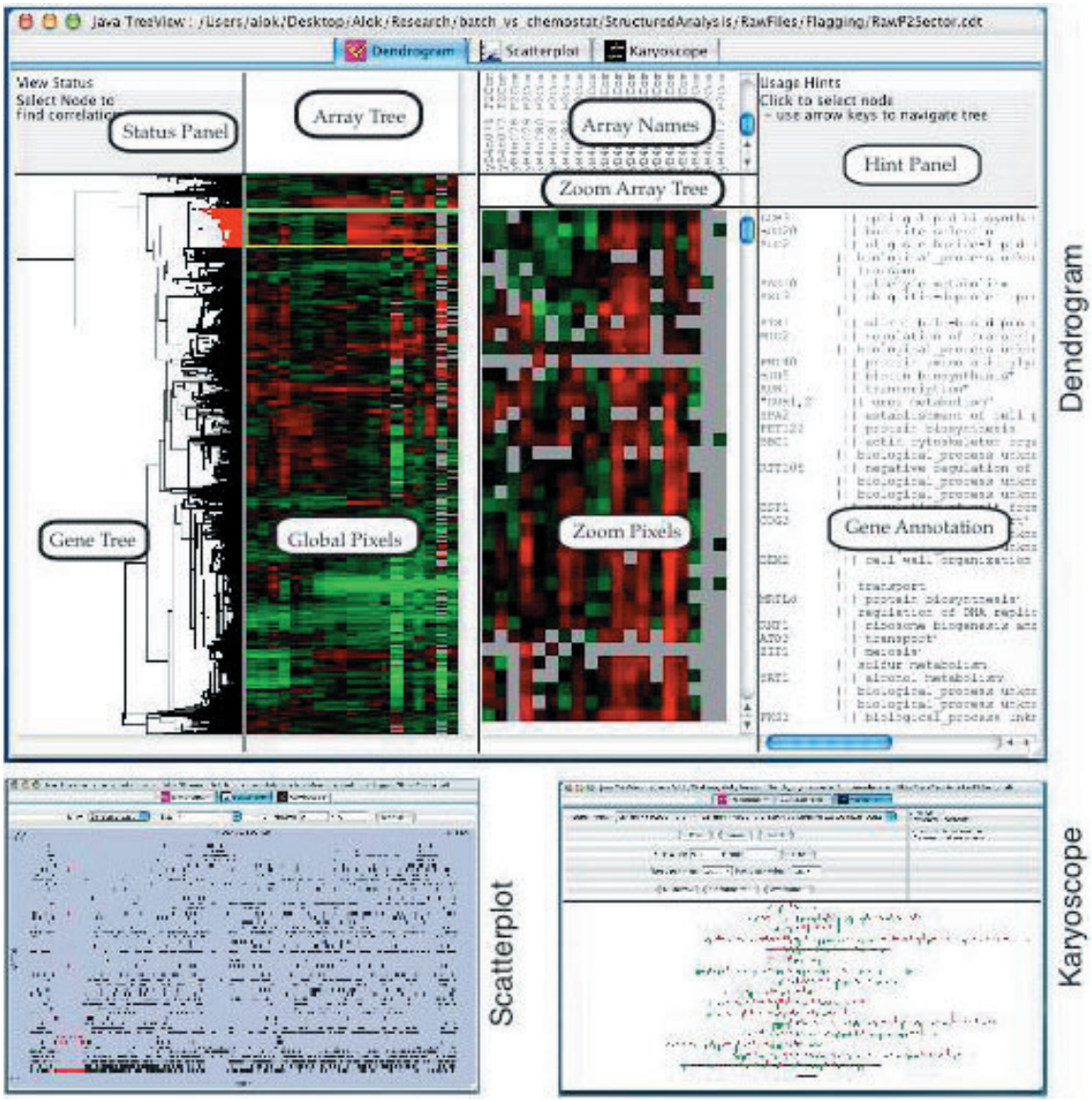

Fig. 1. Multiple displays with linked selections can be produced from the same extended CDT file. In this example, a dendrogram view showing hierarchical clustering of the genes, a scatterplot of sectors on the y-axis against genes on the x-axis, and a karyoscope view of the first array were produced. Genes selected according to the hierarchical clustering in the dendrogram display are drawn in red in the scatterplot display, and appear to come from a small set of sectors, suggesting that this cluster is a spatial bias artifact.

of 233 arrays × 5380 genes = 1 253 540 cells consumes 125 MB of RAM; displaying one of 183 arrays × 21 819 genes = 3 992 877 cells consumes 300 MB of RAM. This puts the display of even very large clusters into the realm of cross-platform computing. Combined with an analysis program, such as XCluster (Sherlock, 2000) (http://genome-www.stanford.edu/~sherlock/cluster .html) or Cluster 3.0 (De Hoon *et al*., 2004), a spreadsheet editor such as Excel, and scripted automation using packages such as PCL Analysis (http://pcl-analysis.sourceforge.net), Java Treeview is a powerful tool for discovering patterns in genome-wide datasets.

BUILT-IN FUNCTIONALITY OF JAVA TREEVIEW

Java Treeview provides fine control over the appearance of the dendrogram. Within the generalized CDT file, the foreground and background colors of gene and array annotations, as well as the color of each node branch in the array and gene dendrograms can be independently specified, greatly increasing the information that can be represented. In addition, time can be used instead of correlation to organize the dendrogram, allowing the visualization of phylogenetic trees. A customizable context-specific information window in the upper left displays information such as the number

of genes selected, the correlation of the selected node and relevant annotations depending upon the cursor location. Instead of a single annotation, arbitrary combinations of annotations can be displayed adjacent to the dendrogram. Clicking on a gene or array annotation opens relevant Web databases in an external browser, provided that there is an annotation column containing an identifier and an URL template for the external database. Templates for SGD(yeast), SOURCE(human), WormBase, FlyBase, the mouse Genome Database and GenomeNet *Escherichia coli* are provided by default, and there is a repository on the website describing additional URL templates.

Java Treeview currently supports three new visualizations in addition to the dendrogram. Arbitrary gene scores can be used to produce scatterplots, gene locations can be used to produce a karyoscope plot and aligned sequence can be used to create a dendrogram-like view of sequence data. Motivating uses of the first two visualizations are provided in the accompanying figure, and sequence alignment visualization is described in the examples section of the website.

Examples, mini-tutorials and an extensive User Manual are available from the website (http://jtreeview.sourceforge.net). An open-source, PERL based package, 'helper-scripts', is provided to aid in the incorporation of data into the .cdt format for visualization.

Currently, all views support output to raster-based images (PNG, PPM and JPEG) and output of subsets of the data to tab-delimited text for further analysis. The dendrogram display also supports output to vector-based postscript files.

ARCHITECTURE OF JAVA TREEVIEW

In order to maximize cross-platform compatibility, Java Treeview is implemented in pure java, using standard swing libraries. The open source, cross-platform Apache Ant is used as the build tool. Although developed on Mac OS X, Java Treeview has been tested on Linux and Windows platforms, and will run on Mac OS9 with the Swing extension. There is also an applet version that enables rich presentation of microarray data on supplementary websites with no additional programming (see examples section of http://jtreeview.sourceforge.net). Settings are stored separately for each CDT file in an automatically generated XML formatted '.jtv' file. The '.jtv' file can be placed on a website with the CDT file to customize many features of the applet, including visualizations, colors and url-linking. Documentation is written in the DocBook XML format, with automated transformation into PDF and HTML. Interested developers should consult the Programmer's Guide available from the main website.

ACKNOWLEDGEMENTS

I would like to thank J. Michael Cherry for his support, David Botstein for allowing me to pursue this project and Michiel Jan Laurens de Hoon for contributing the Windows installer. This work was supported by a National Defense Science and Engineering Graduate Fellowship, the Stanford Genome Training Program (Training Grant NIH 5 T32 HG00044) and NIH grants HG 01315 and GM 46406.

REFERENCES

De Hoon,M.J., Imoto,S., Nolan,J. and Miyano,S. (2004) Open source clustering software. *Bioinformatics*, **20**, 1453–1454.

Dysvik,B. and Jonassen,I. (2001) J-Express: exploring gene expression data using Java. *Bioinformatics*, **17**, 369–370.

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci., USA*, **95**, 14863–14868.

Johnson,J.E., Stromvik,M.V., Silverstein,K.A., Crow,J.A., Shoop,E. and Retzel,E.F. (2003) TableView: portable genomic data visualization. *Bioinformatics*, **19**, 1292–1293.

Krull,M., Voss,N., Choi,C., Pistor,S., Potapov,A. and Wingender,E. (2003) TRANSPATH: an integrated database on signal transduction and a tool for array analysis. *Nucleic Acids Res.*, **31**, 97–100.

Sherlock,G. (2000) Analysis of large-scale gene expression data. *Curr. Opin. Immunol.*, **12**, 201–205.

Sturn,A., Quackenbush,J. and Trajanoski,Z. (2002) Genesis: cluster analysis of microarray data. *Bioinformatics*, **18**, 207–208.

Zhou,Y. and Liu,J. (2003) AVA: visual analysis of gene expression microarray data. *Bioinformatics*, **19**, 293–294.