A browser for expression data

M. R. Pocock* and T. J. P. Hubbard

The Sanger Centre, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SA, UK

Received on July 16, 1999; revised on September 27, 1999; accepted on October 12, 1999

Abstract
Summary: We have written a fully extensible Java application for visually browsing expression data, and clusters of genes or experimental conditions calculated from that data. The application requires a runtime environment for Java2.

Availability: http://www.sanger.ac.uk/Users/mrp/java/ExpressionBrowser
Contact: mrp@sanger.ac.uk

Introduction

More and more expression data is being collected systematically using methods such as quantitative PCR (Delidow et al., 1989; Wang et al., 1989), array technologies (DeRisi et al., 1997; Schena et al., 1995) and DNA chips (Johnston, 1998). Often, this data is analysed using cluster algorithm. These find groups of genes with similar expression profiles, and groups of conditions that have similar effects upon the genes [for example see Wen et al. (1998)]. As data set become larger, it becomes less possible to view all of the raw data simultaneously. However, it is essential to keep an eye upon the structure of the whole data set when choosing which subsets to investigate further.

Our Expression Browser allows the user to view the results of external clustering programs as trees, and invoke actions upon sub-trees or terminal labels using the mouse. In addition, the expression data for the selected genes and conditions are displayed in another window so that the user can drill down to the information of interest.

The application is independent of the means used to store the expression data, and the cluster analysis methods used. For information publishers, there is a clear and extensible framework for publishing expression data, and for adding capabilities to the browser without modifying or recompiling any of its source code. It uses XML, URLs and JAR files to package and locate both code and experiment information in a modularly extensible manner, while retaining the tight type checking of both validated XML and Java code. If used in conjunction with certification and encryption, it is also a secure solution for data dissemination over the web.

The Expression Browser

The browser is written using Java 1.2, and relies upon the Java Project X libraries for processing XML files. Java 1.2 was chosen because it provides good mappings to SQL and CORBA and includes the Swing and 2D graphics libraries. The browser was developed specifically as an application to avoid some of the security and compatibility issues associated with developing applets that rely upon new Java language features.

When the browser is launched it displays the main window. Experimental sets are loaded via the File menu from URLs, which may be local file URLs, or any other protocol supported by Java, such as http or ftp. Once the experimental file is loaded, the user can select the trees that they wish to view from the gene or condition list.

The trees are pre-computed and stored as files containing the results of any cluster analysis program in a parseable format. We have generated clusters using SOTA (Wang et al., 1998), as well as our own clustering software (in preparation), and represent the clusters in the default format of an XML document containing elements describing the tree. Newick formatted trees will be supported in the near future and other bindings will be developed as requested.

The trees display show the results of a pre-computed cluster analysis as a tree. The visual layout of the tree is controlled by the ‘Tree Builder’ selection. For example, there are tree builders for horizontal trees and for circular, radial distance trees. Next to the ‘Tree Builder’ selection box are two other selection boxes that associate mouse actions with tools. One set of tools is named ‘Clade tool’ and the other is ‘Leaf Tool’. The clade tool is activated when the user clicks within the tree, selecting a clade or sub tree. The leaf tool is activated when the user clicks on a label of the tree. The core distribution provides the following tools:

- selecting sub trees and leaves
- browsing a sub tree in a new window or replacing the
current tree with a sub tree

- retrieving the expression data for a set of genes and displaying it as a grid of coloured boxes
- displaying the EMBL (Stoesser et al., 1999) entry for a gene using SRS (Etzol and Argos, 1993).
- loading an arbitrary URL, such as an image.

In addition to these tools, the author of the experiment file can add experimental specific tools. For example, gel images could be retrieved for each gene measured in a quantitative PCR data set, or the image of a DNA micro-array shown when the micro-array is clicked in the condition tree window. In this way, information is made available about genes or conditions.

The leaves of a tree can be searched using a dialog box. Selecting ‘Find → find the leaf’ from the menu makes a ‘Find leaf’ dialog box visible. From here, the user can click on leaf names listed and see them highlighted in the tree. Entering a name fragment into the top text box restricts the leaves shown to just those containing that string.

Each experiment has an associated ‘Selection’ window. This can be used to view the expression data for a subset of genes and of conditions. The subsets can be set to one of two behaviours independently for genes and conditions:

- All   all the items will be selected all the time
- Sync  the subset of items selected is tied to the selection within a tree window

For example, the gene selection may be set to synchronize with the selection within a gene tree window, while all the conditions are displayed.

In addition, the genes or conditions can be quickly clustered using an in-built cluster algorithm so that they are recorded to give a clear visual impact. When looking at a single clade, this shows why the original cluster algorithm put the items together on the tree. By combining selections from different parts of the trees, it is possible to get a very clear idea for the reasons why the two clades where placed apart.

The colour schemes for displaying expression profiles are totally configurable. For example, discrete data may be coloured using a colour scheme that gives clearly different colours for each category, while continuous data may be plotted as a range of colours. This potentially makes patterns in the expression data much easier to visualize.

**Conclusion**

The Expression Browser application is a good solution for publishing expression data and clusters of that data in an interactive form. It also provides rich opportunities for extending the basic functionality to incorporate data-set specific information, such as images or links to external databases.

**References**


