Reusable Graphical Interface to Genome Information Resources

Andrei Grigoriev

Max-Planck-Institute for Molecular Genetics,
Ihnestr. 73, 14195 Berlin-Dahlem, Germany

This paper describes a prototype genome display and query system for the World Wide Web, which could play the role of a graphical interactive gateway to online genome information services. It provides a uniform interface to display mapping and sequencing data for the human, mouse and yeast genomes and could be easily extended to accommodate more information as it becomes available. This system uses a Java applet, DerBrowser, for delivering interactive content to an end user. The architecture and functionality of this applet are described, with respect to views of both users and data providers.

1 Introduction

In genomic mapping and sequencing, flexible visualization tools are very important for monitoring experimental progress, locating inconsistencies in the data and presenting the final result to the scientific community. A well-known example of such software tool is QUICKMAP, distributed alongside the CEPH-Genethon YAC libraries[1]. Since this software runs on a limited number of platforms, its recent port to the World-Wide-Web (WWW) has allowed more users to access experimental results obtained with these clone libraries (although currently without a graphical interface). A large number of other genomic research centers are providing similar online information services on several genomes (for example, a comprehensive list of online resources on the human genome is available at http://www.hgmp.mrc.ac.uk/Public/human-gen-db.html).

Arising from independent efforts, these services utilize different data models, database management systems and server software to disseminate information over the WWW. While collating the data available from such diverse sources into the most consistent overall maps remains a challenging task for bioinformatics, a more immediate problem of providing an experimentalist with an intuitive graphical interface, displaying a map and, importantly, allowing one to retrieve up-to-date information about mapped objects from all sources accessible online, could be addressed today.

This paper describes such a prototype reusable display/query interface for various types of mapping and sequencing data which could be used in different genomic projects without any modification.

2 WWW Interfaces

The Internet and WWW are quickly becoming the mainstream for dissemination of up-to-date biological information. Hypertext Markup Language (HTML) [10], imagemaps and Common Gateway Interface (CGI) [8] are currently widely used in WWW interfaces to online databases and other computational resources. With the arrival of Java language developed at Sun Microsystems [4], these static and bandwidth-consuming approaches can be replaced by platform-independent client-side interfaces.

A number of interactive map display tools written in Java have emerged in the past few months (an interesting recent collaborative review is available at http://www.cherwell.com/javagenomes/). While varying in actual interactivity and display capabilities, most of them are tightly coupled with underlying project/database data models and may require significant changes when reused in other genomic projects.

However, the linear representation of mapping and sequencing data, being very similar for various types of maps, suggests the possibility of a flexible reusable interface for their output. Such prototype interface presenting data on different genomes in a standard interactive form is a basis of Genome Navigator [6], a genome display/query system for the WWW, which could play the role of graphical interactive gateway to the major genome information servers. In essence, this is an attempt to link a map position to the corresponding functional and structural data by a few mouse clicks. Currently, Genome Navigator serves maps of the human, mouse and yeast genomes but could be easily extended to accommodate more information as it becomes available.

Genome Navigator employs DerBrowser (a Java applet, http://www.mpimg-berlin-dahlem.mpg.de/~andy/DerBrowser) as its display tool and CGI scripts to serve data to the applet. Currently, a static local data copy is used for each genome but nothing prevents a data provider from collecting data dynamically from external sources, avoiding Java security restrictions via server-side CGI scripts.

The applet provides capabilities to query external databases via their own (existing) CGI scripts. The growing list of data sources used by Genome Navigator currently includes: Whitehead/MIT, GDB, CEPH-Genethon Infoclone, CHLC, Human Transcript Map, and several human chromosome-specific databases; EUCIB, Saccharomyces Genome Database, YPD, MIPS and GeneQuiz (for an up-to-date list of references and links to these servers, see the Genome Navigator page [6]). The database management systems of these servers are very different (Oracle, Sybase, ACEDB, etc) and the applet utilizes their ability to provide a CGI-based queries by object name. In the

future, this may be replaced by other architecture standards of server-client communication, like CORBA.

3 Applications

The main application of DerBrowser is to serve as a WWW interface for a wide range of data sources. Its primary use is probably with genomic databases (and originally it was developed as a front end to the IXDB, the database of the mapping information on the human X chromosome [9]), while one could envisage it serving online journal map illustrations, with all the added benefits of interactivity and connections to other servers.

Another important application would be as an interface component to represent the output of WWW-based data analysis tools. One example of such use is the physical mapping server on the WWW site of MPIMG [5], where the applet displays physical maps constructed from uploaded raw hybridization data.

In addition to the WWW interface capabilities, the same applet could be employed as a front end of local data analysis software. Instead of spending time on a graphical interface, a developer can quickly append an output in specified format to an existing or newly developed program, and simply load such generated local HTML file into Netscape or another Java-capable browser to view results in graphical form on any hardware platform with the help of the applet. For example, the next release of the contig-building software package [7] developed at ICRF and MPIMG, includes physical map and inter-contig connection displays using DerBrowser, with links to the original program output converted into hypertext files.

Another useful application may be in including the applet as a viewing tool into regular map/sequence distributions on CD-ROMs.

4 Functionality

In principle, one could consider DerBrowser as a Web publishing tool for molecuar biologists working with mapping and sequencing data. Hence, there is a clear distinction between users and data providers so the functions of DerBrowser are best described separately for these two groups.

4.1 User's prospective

The purpose in this case is to bring a familiar, compact and clear image of a map/sequence to a user, allowing for customization and maximizing informa-

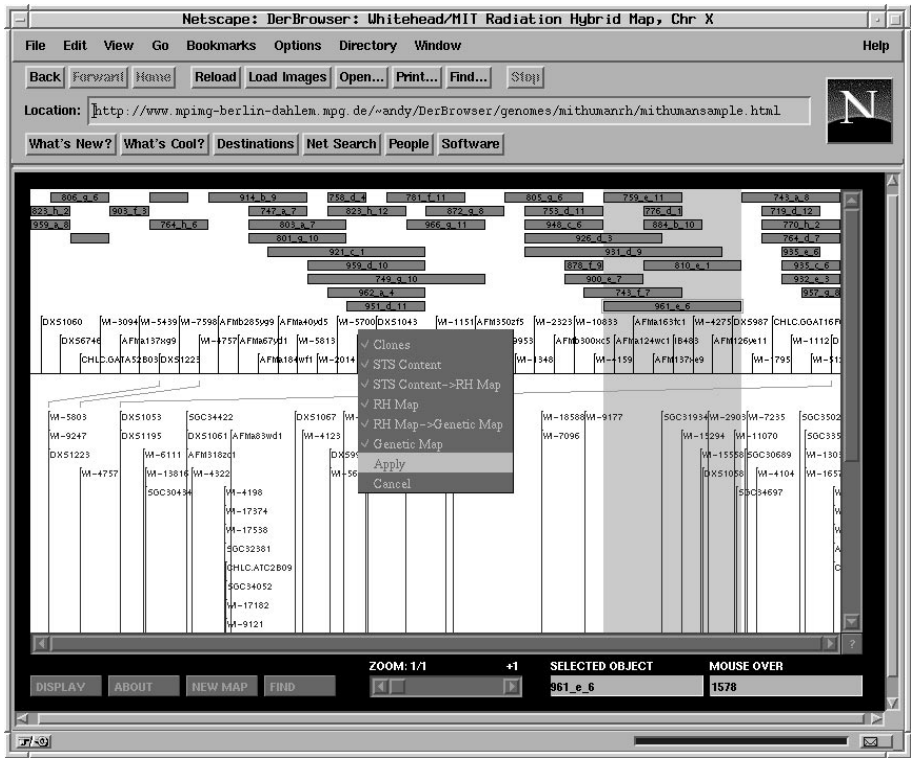

Figure 1: Screenshot: DerBrowser displaying human X chromosome data, with the menu of available object groups (stripes).

tion output at every mouse move and click.

The user is presented with a horizontal display of related groups of linear objects, with a scale at the very bottom. Colored rectangles are used to display clones, chromosome bands, sequences, etc, while vertical bars are typically markers, whose names are always visible. The user can zoom into the map and, if space permits, names of objects shown as rectangles will become visible. Otherwise, moving a mouse cursor over an object will display this object's name and group in the MOUSE OVER window (if no object is under the cursor, a map coordinate of the cursor is shown instead). Horizontal and vertical scrollbars are also provided for easy navigation.

Pressing the DISPLAY button, the user can switch off and on the display of the available groups of objects, listed in a dynamically generated pop-up

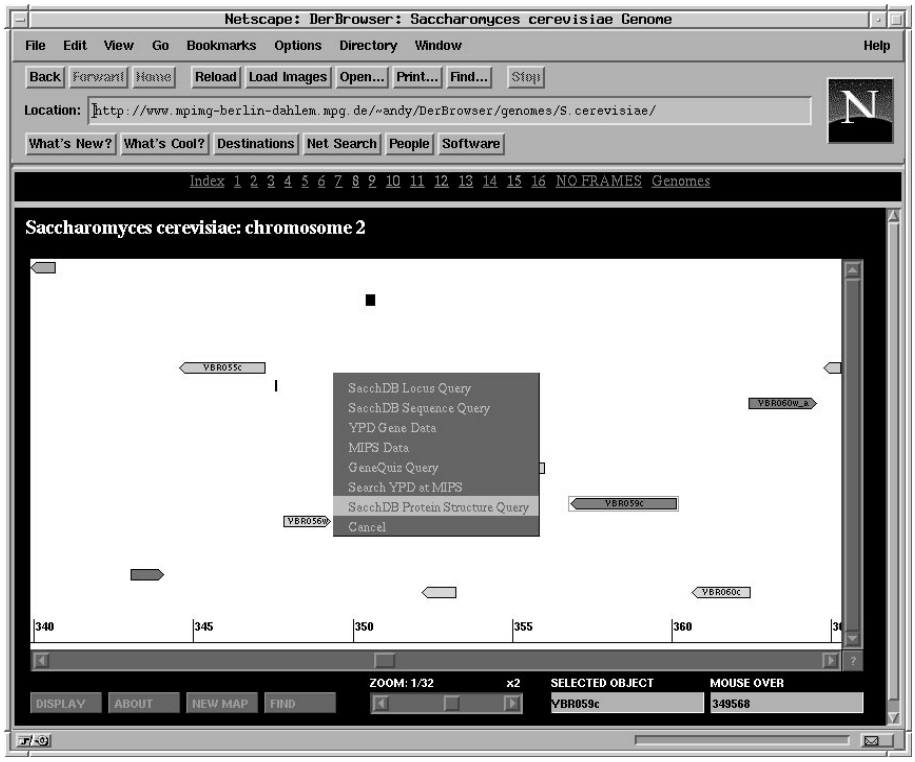

Figure 2: Screenshot: DerBrowser displaying S. cerevisiae data, with the menu of available external data sources).

menu (Fig. 1). Additional customisation (fonts, zoom step and other features) is available via the "?" (question mark) button at the intersection of the scroll-bars.

If provided, a new map could be retrieved using the NEW MAP button and selecting map boundaries and object groups of interest.

An object could be found on the map by pressing the FIND button and entering a part of the object's name. If more than one object is found, another dynamically generated pop-up menu will present a list of names of all found entities. Clicking on any of these names will select the corresponding object and place it in the center of the display.

A selected object is marked by a red border around it, its name also shown in the SELECTED OBJECT window. Upon selection, the user may

press the ABOUT button (Fig. 2) to find a menu of provided external data sources, which will bring more information (if available) about the selected object in a separate browser window. Queries to such external data sources are performed transparently for the user, who doesn't have to know anything about the underlying mechanism, query structure, etc.

4.2 Data provider's prospective

In this case the goal is to give a data provider maximum flexibility without the need to rewrite Java code, modify database or existing server functionality. What follows is a brief description of the interface architecture and its capabilities.

The more sophisticated the data model incorporated in an interface is, the less flexible and reusable the interface itself becomes. Given the great diversity of currently available data sources, the map display interface was developed without any rigid biological data model behind it. Only the information necessary to display objects on the map is passed to the applet, including object's name, an arbitrary number of IDs (for querying external data sources), coordinates and color. These mapped objects are called "elements" and the data provider supplies them in groups of one's own choice (e.g., YAC clones, cosmid clones, markers, etc.). Such groups constitute independent parts of a display, laid out as horizontal "stripes".

In the current implementation, the applet operate with stripes and propagate events (e.g., mouse clicks) to them; stripes represent various object types, hold, draw and manipulate elements. Elements (e.g., individual clones) only hold values supplied by the data provider. To achieve that, the *Stripe* class has the following methods

- Arrange elements to avoid overlap
- Calculate stripe size
- Draw itself and its elements
- Find element by name
- Find element by coordinate

All stripe types extend the same *Stripe* class. Each subclass of *Stripe*, representing one of the types of map objects, handles its elements differently and provides its own drawing, rearrangement and other methods. For the applet, actual stripe types are defined when reading data and then all stripes are accessed via generic vector

`stripe=(Stripe)stripeVector.elementAt(x)`

This architecture gives great flexibility for displaying various types of maps, from a level of chromosome bands down to sequence. For a developer, it also means that new stripe types can be added easily and any existing type can be modified or replaced without affecting functionality of others.

Currently, the object types that can be displayed by the applet include *Multi-locus, Clone, Locus, Multi-map, Plot, Exon-Intron* and *Sequence*. Their differences and examples of use are listed in Table 1.

Data presentation and display functionality are controlled by a data provider via parameters (applet tags) in the HTML file containing the applet. The applet takes a number of parameters following the HTML 3.2 and Java applet tag specifications:

- LEFTEND/RIGHTEND
- USERID/MAPID
- TYPES/OTHERTYPES
- PORT
- SCALEUNIT
- NOCOORD
- CGI/FILE/CGIENCODE
- MOREx/MORETEXTx/MOREENCODEx.

The meanings of the tags are described below.

Positional data on mapped objects are sent to the applet via a data stream in a standard format (relational table). If the data stream is a flat file, then its filename is specified in the FILE tag. Otherwise, a CGI script (called by the applet) provides the data - then a call to this script should be given in the CGI tag. The latter allows for variable call format (see 4.3). A server PORT can also be specified.

Normally, a certain user (USERID) would want to see a certain segment (from LEFTEND to RIGHTEND) of a certain map (MAPID). The SCALEUNIT tag determines the numbers shown on the scale (default 1000, for kilobase scale for data given in basepair values). The map position of the mouse cursor is normally available in the MOUSE OVER window - this can be switched off using NOCOORD tag.

The provider may also specify the map stripes/elements to be shown. This is passed to the applet in TYPES tag as a comma-separated list (eg

"YAC,COSMID,Genethon Markers"). The order of stripe names in this tag is also used to specify the order of stripes on display (top to bottom).

Other map stripes, present in a database but not shown this time, can be specified as OTHERTYPES (also a comma-separated list). They will be used to retrieve a new map with these stripes included, if the user so wishes.

The user may request extra information about selected elements. This

Table 1: Description of available stripe types.

Stripe	Elements overlap	Elements rearranged	Names always visible	Examples
Multi-locus	No	No	No	chromosomal band, radiation hybrid, restriction fragments
Clone	Yes	Yes	No	clones, from YAC to plasmid, overlapping contigs
Locus	Yes	Yes	Yes	loci, genes, markers, STS, etc., custom scale
Multi-map	Yes	No	No	comparative, integrated genetic and physical maps
Plot	Yes	No	No	graphs and histograms along the map
Exon-Intron	Yes	Yes	No	exon-intron structures
Sequence	No	No	No	nucleotide or protein sequence

could come from a number of sources. Each of this sources is a CGI script or a file. They can be specified as MORE0, MORE1, MORE2,... tags, allowing for variable call format (see 4.3). Textual explanation of these sources should be given in the corresponding MORETEXT0, MORETEXT1, MORETEXT2,... tags. The user sees these explanations as a dynamically generated menu of additional queries about selected objects. For each of these queries, tags MOREENCODE0, MOREENCODE1, MOREENCODE2,... specify if a corresponding variable call should contain URLencoded data (if data contains symbols like spaces, slashes, etc, which normally are encoded as CGI script parameters). Analogous CGIENCODE tag is available for the same purposes if a CGI call is used to get the data stream.

If just one tag - MORE0 - is given, then ABOUT button will be linked to only one call. If more sources of additional information are specified, a menu of those (designated by MORETEXTx values) will be presented to the user.

4.3 Variable call format

DerBrowser is an attempt to provide a uniform interface for various data sources. Many of these sources are databases with their own ways of serving data on the WWW, and it will often be useful to poll more than one data source for more information on a selected element. Variable call format is a flexible way to include user selection context parameters into database-specific CGI scripts.

The following format entries will be substituted (and URLencoded, if necessary) by their values:

- MAPID - current map
- USERID - current user
- LEFTEND - left end as tag value or start of the new map given by user
- RIGHTEND - same for the right end
- TYPES - list of stripes as tag value or selection for the new map requested by user
- OBJID - selected object id
- OBJNAME - selected object name
- STRIPEID - selected object's stripe id
- STRIPENAME - selected object's stripe name

These entries should be surrounded by spaces, which are removed from the translated calls. Spaces have been chosen simply because these symbols should not be present in fully qualified URLs or URLencoded strings.

For example, the following CGI tag

```
<PARAM NAME="CGI" value="/cgi-bin/mapdata.cgi?map= MAPID &user
= USERID &left= LEFTEND &right= RIGHTEND &show= TYPES">
```

will result in a server call like

```
http://server:port/cgi-bin/mapdata.cgi?map=1&user=someuser&left=
0&right=10000&show=YAC,Cosmid,Marker
```

Or the following combination of MORE0/MORETEXT0 tags

```
<PARAM NAME="MORE0" value=" http://rag2.rz-berlin.mpg.de:8888/
ixdbbin/more.cgi?mapid= MAPID &userid= USERID &objid= OBJID">
<PARAM NAME=" MORETEXT0 " value="IXDB">
```

will place "IXDB" as the first item in the menu of additional queries about selected objects, and on selecting this item will generate a CGI call to the IXDB database regarding the object number 1679 as

```
http://rag2.rz-berlin.mpg.de:8888/ixdbbin/more.cgi?mapid=1&
userid=01&objid=1679
```

This simple and straightforward format gives a data provider sufficient power and flexibility to feed a desired subset of mapping data into the applet or form a variety of queries to any available external data sources. A similar, although more limited, context-dependent applet action has been described in Felciano [3].

4.4 Access restrictions

Often, the same database contains publicly available information together with data access to which is restricted to limited (sometimes several different) groups of users. A combination of USERID and MAPID parameters of the applet (generated by CGI scripts in protected server areas) could be used to maintain different access restrictions, as done in IXDB [9].

5 Future developments

The new version of the Java Development Kit (Sun Microsystems) provides changes in the applet security model which will help in improving the applet usability by permitting it to save files on user's local disk and access printer. Corresponding functional changes (edit/save and print a map) are currently underway and will be available in the new release of DerBrowser. Additionally,

a modification to enable simple connection to the ACE server architecture is planned, which would allow to use the applet as a network display tool for the ACEDB-based databases[2].

6 Conclusions

The platform-independent graphical interface presented here is a reusable solution for interactive presentation of linear genomic (and other) data. It is designed to work in conjunction with other distributed and local data sources and analysis software providing them with user interaction context. It allows data providers to construct flexible map displays linked to online information without source code level programming. It will be made freely available to the academic community in the nearest future.

Acknowledgments

This work was in part supported by the grant 01KW 9608 of the BMBF Human Genome Project. I would like to thank Leonard Schalkwyk for critical reading of the manuscript.

References

1. Chumakov., I. *et al*, (1995) A YAC contig map of the human genome. Nature, Genome Directory, vol 377 SUPP., September 28, 1995, pp 174-297. See also http://www.cephb.fr/quickmap.html
2. Durbin, R., and Thierry-Mieg, J. The ACEDB Genome Database. Documentation available at http://probe.nalusda.gov:8000/acedocs/
3. Felciano, R., Chen, R., and Altman, R. (1997) RNA Secondary Structure as a Reusable Interface to Biological Information Resources. (to appear in Gene-COMBIS).
4. J. Gosling and H. McGilton, 1995. The Java Language Environment. Sun Microsystems White Paper (http://java.sun.com/whitePaper/java-whitepaper-1.html)
5. Grigoriev, A., Levin, A. and Lehrach, H. (1997) Distributed Computing in Genomic Mapping: Building Contigs on the Web. German Conference on Bioinformatics - 97, Kloster Irsee, Germany. (http://www.mpimg-berlin-dahlem.mpg.de/~andy/server/).
6. Grigoriev, A. (1997) Genome Navigator, http://www.mpimg-berlin-dahlem.mpg.de/~andy/GN/).

7. Mott, R. F., Grigoriev, A. V., Maier, E., Hoheisel, J. D., and Lehrach, H. (1993) Algorithms and software tools for ordering clone libraries: application to the mapping of the genome of Schizosaccharomyces pombe, Nucleic Acids Research 21, 1965-1974.
8. National Center for Supercomputing Applications (1995) The Common Gateway Interface, http://hoohoo.ncsa.uiuc.edu/cgi
9. Roest Crollius, H., and Leser, U. (1996) An Integrated X chromosome database (http://www.mpimg-berlin-dahlem.mpg.de/~xteam).
10. World Wide Web Consortium (1996) HyperText Markup Language (HTML), http://www.w3.org/pub/WWW/MarkUp