EPCC-SS98-11

JavaGene: Java Middleware for Integrated Access to Distributed Molecular Biology Resources

Scott Trudeau

Abstract

Research in molecular biology is among the most exciting areas of investigation today. Developments are occurring rapidly. Among the factors that have accelerated these developments are computer based research tools. A wide variety of tools are used for storing, finding, and analysing genetic data. Using many tools on different platforms can cause frustrations even for users experienced with the variety of tools and operating systems. For research biologists with limited computer experience, their ability to use the available tools effectively becomes limited. The goal of JavaGene is to provide integrated access to available tools for all potential users.

Contents

1 Background **3**
1.1 Molecular Biology Research 3
1.2 Typical Process 3
1.3 Problems With This Process 3
1.4 Typical User 3

2 Common Research Tools **4**
2.1 Sequence Storage 4
2.2 Finding Sequences 4
2.3 Analysing Sequences 4

3 Goal **4**

4 Design **5**
4.1 User Interface 5
4.2 File Operations 5
4.3 Sequence Retrieval System 5
4.4 Genetics Computing Group Package 6
4.5 Local Perl Scripts 6

5 Conclusion **6**

6 Future Development **7**

A JavaGene Classes **8**
A.1 ProxyServer 8
A.1.1 Configuration File 8
A.1.2 Security Mechanisms 8
A.1.3 HTTP and HTML Communications 8
A.2 JavaGeneServer Class 8
A.2.1 User Interface Model 8
A.3 Utilities Class 9
A.4 TextFile Class 9
A.5 gcgServer Class 9

1 Background

1.1 Molecular Biology Research

Deoxyribonucleic acid, or DNA, is the material of heredity, and as such is of principal interest to researchers in genetics and molecular biology. DNA is a polymer composed of four simple components termed bases. Each chromosome contains a pair of DNA strands, each of which contains of order 10^8 bases in mammals. The sequence of bases in DNA serves as a genetic blueprint, storing all the information needed for each organism to survive and reproduce [4]

A gene is a specific DNA segment that encodes the information needed by the cell to produce a particular protein which then contributes to some function in the organism. Genes are usually what researchers are trying to isolate, analyse, and compare when looking at data in DNA sequences. Humans are estimated to have about 80,000 genes. To date, only about 5,000 genes have been fully sequenced. However, with the Human Genome Project on schedule to determine the entire 3 billion bases in the human genome by 2005, the amount of DNA sequence in the public databanks is growing exponentially. [5]

Computational tools are increasingly necessary for lab biologists to effectively utilise this explosion of data.

1.2 Typical Process

Using currently available tools to pursue research can be frustrating. It requires users to have experience with a number of different tools.

A researcher first discovers a new, possibly interesting sequence using laboratory techniques. This sequence is then stored on the researcher's own PC. The sequence then must be uploaded using FTP to a local Unix machine where local scripts can be used to analyse, edit, or convert the sequence. The user then may seek similar sequences from data repositories on the web for comparison. Then these sequences may be sent over to another central Unix machine for further analysis using other, proprietary tools. The results from this analysis must then be sent back to the local Unix machine for further analysis. Furthermore, the user may then wish to bring some of the resulting data back to their own PC.

1.3 Problems With This Process

This process requires users to have a certain body of knowledge. They must keep track of files across at least three separate platforms. They must have a working knowledge of using the Unix command line and tools like FTP. Unfortunately, most potential users lack this knowledge.

1.4 Typical User

The typical user is most familiar and comfortable with Macintosh PCs. They are comfortable using graphical tools like web browsers and word processors. They have little to no experience working in Unix and other environments. Thus, typical users are unable to use all available tools effectively, if at all. Even experienced users are frustrated by the hassles involved.

2 Common Research Tools

As previously mentioned, a wide variety of computer based tools for molecular biology research now exist. The three types of computer based tools researchers use when studying DNA sequences are used to store, find, and process and analyse sequences.

2.1 Sequence Storage

Typically, users store sequences in files on their PCs. Keeping track of different incarnations of the various sequences in different file formats is often a hassle. When they find a sequence that may be of interest to the global research community, they submit it to remote data repositories making their work available to the wider community.

2.2 Finding Sequences

One of the most powerful tools available for finding interesting sequences is the Sequence Retrieval System (SRS) managed by the European Bioinformatics Institute (EBI). Their web site [1] provides a form based front end for searching over 100 separate databanks of genetic information.

2.3 Analysing Sequences

To analyse data, researchers in the Institute of Cell and Molecular Biology use a variety of tools:

- The proprietary Genetics Computer Group package (GCG), a powerful collection of over 100 tools for sequence analysis. Being an expensive proprietary package, GCG is only available on a central Unix system (holyrood).
- A collection of Perl and shell scripts executable through an X-windows file manager running on a local Unix workstation.
- Various proprietary packages on local PCs.
- Web-server utilities, accessed through browsers on local PCs.

3 Goal

Ideally, users would like to be able to use all of these available tools without having to learn a great deal about operating systems, file transfers, and specific commands.

JavaGene is an attempt to provide integrated access to these distributed resources with a user-friendly interface, accessible to all potential users.

4 Design

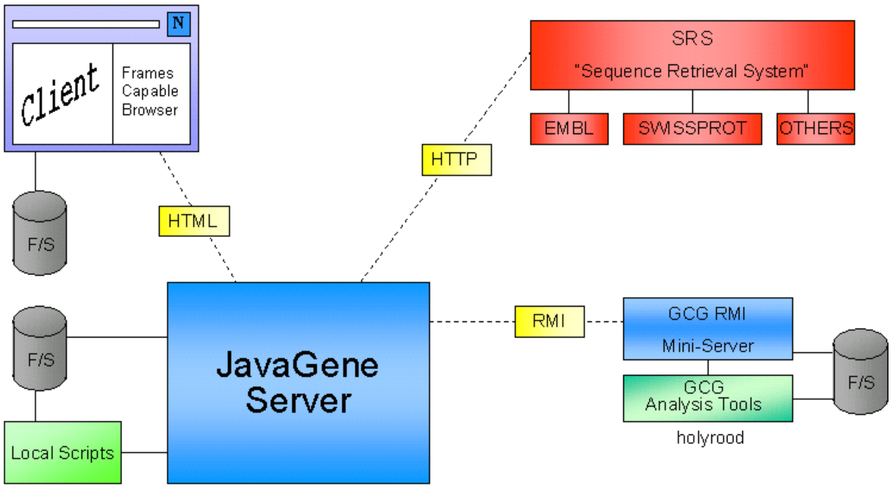

Figure 1: Component Architecture

4.1 User Interface

The JavaGene server sends all output to the user in frames-based HTML and gathers all input via HTML forms. This allows any user with a frames-capable browser to access the tools available through JavaGene. The interface is divided into four main frames. There is a small title frame; a frame containing menu options; a frame displaying the current folder and its contents along with some file operation options; and the main frame for user input and output.

4.2 File Operations

One of the main benefits of JavaGene is that it provides a simple, graphical interface for managing files on the machine hosting the JavaGene Server. It uses the Java File class for these operations. It allows users to create and remove folders; delete, rename, upload, download, and view text files. All the files the user sees are located and managed on the host server. Also, specialised icons have been designed to make files with common extensions easy to recognize. Keeping track of files across systems and knowledge of Unix commands is no longer necessary.

4.3 Sequence Retrieval System

The European Bioinformatics Institute provides a relatively robust web based front end to SRS. JavaGene simply acts as a proxy server for the SRS web page. It intercepts HTTP request from the browser, forwards them off to the SRS web server, and brings the response back for display. This allows users to use the existing, familiar interface. There is one catch. The output SRS gives when saving a sequence is in an undesirable form. In order to save sequence files to the JavaGene server and get around this quirk in the saving process of SRS, JavaGene includes its own save function. When a user finds a sequence they can then choose to save it to JavaGene, and JavaGene will handle the saving and formatting of the file.

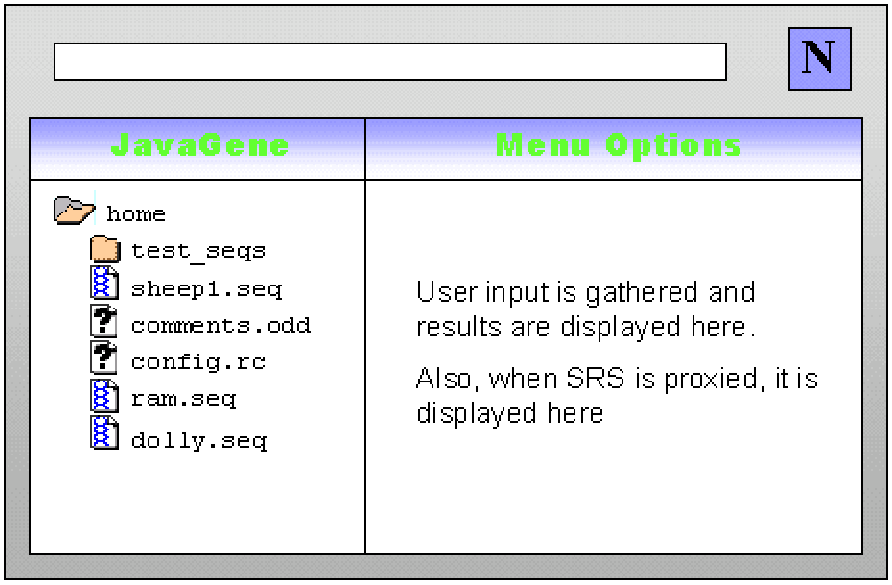

Figure 2: User Interface

4.4 Genetics Computing Group Package

The Genetics Computer Group package is a powerful collection of many sequence analysis and visualisation tools. Due to licensing expense installing it on a more convenient machine is not an option. JavaGene uses the Java Remote Method Invocation API (RMI) to make calls to a small server running on the machine containing the GCG executables. JavaGene Server gathers relevant input from the users, packages up any input file(s) and sends the request to the GCG server in a method call. The GCG server writes the input file(s) to disk and executes the command. It then sends the output file back to the JavaGene server where it is written to disk and appears in the file listing along with the results of the operation. The three GCG functions implemented so far — Pileup, Mapplot, and Reverse — are amongst those most frequently needed by users.

4.5 Local Perl Scripts

Also available to users are locally developed Perl scripts that handle simple analysis and formatting tasks. JavaGene provides an interface similar to GCG for choosing and setting up scripts. It uses java.lang.Runtime and Process to execute the scripts.

5 Conclusion

Using JavaGene a user can do what took hundreds of keystrokes at a Unix command line previously with a few clicks of a mouse. Using these tools no longer requires knowledge of Unix or other operating systems and tools. The only knowledge required is how to use a web browser. All resources are presented in a common, easy to use and understand interface.

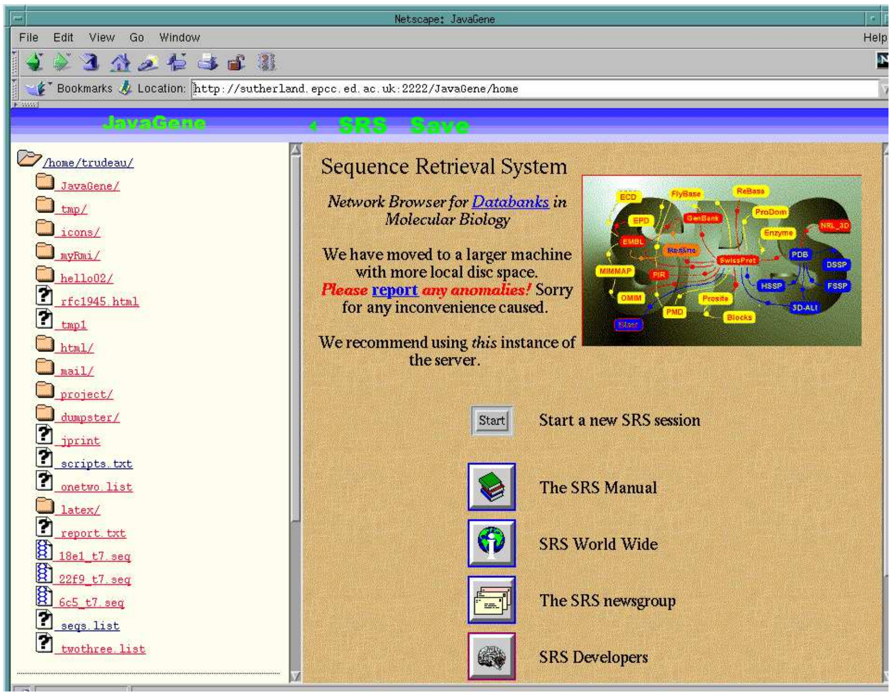

Figure 3: Screenshot

The challenge of providing integrated access to distributed databases and software tools on heterogeneous systems also arises in many other domains outside molecular biology. The strategy used in designing JavaGene may prove applicable in these other areas.

6 Future Development

JavaGene, like software, has room for improvement. Currently, only three of the many available GCG commands are implemented, and only one script is implemented. The way in which the program is coded, however, does allow for easy expansion and customisation of the code. There are still a number of features that could be added or improved in JavaGene. A wider array of GCG tools and local scripts would be valuable. A user maintained database to allow for the storing and retrieving of sequences would be useful for sequence storage management. Interfaces could be added to other search and analysis tools on the web. In particular, an interface to the public-domain CINEMA Java Applet which allows graphical alignment of multiple sequences, would be desirable.[7]

Appendices

A JavaGene Classes

A.1 ProxyServer

ProxyServer was written mainly by Dr. Hon W. Yau of the EPCC. It is the root class of the JavaGeneServer. It provides methods for implementing a configuration file, secure access, HTTP communications with the client and remote URLs, simple HTML replies to the client, and the ability to run processes on the host.

A.1.1 Configuration File

ProxyServer reads in a configuration file and stores specified options into a hash table. This allows for easy extensability of the class, because any number of new options can be added to the configuration file without having to modify the ProxyServer code.

A.1.2 Security Mechanisms

ProxyServer also has some rudimentary security mechanisms. It allows the user to turn any or all of these mechanisms on and off. It provides a primitive user/password authentication, the ability to restrict access to a limited set of IPs, and the ability to allow only the first IP making a request, to continue making requests.

A.1.3 HTTP and HTML Communications

ProxyServer provides the ability to reply to the client with a local or remote URL, and to construct and reply with valid HTTP headers. It also provides methods for writing standard HTML headers and tails to the client. These methods are the heart of the proxy server and make it easy to implement the HTML based user interface.

A.2 JavaGeneServer Class

JavaGene extends the ProxyServer class. It parses each URL, looks for, and then handles, recognised commands. The JavaGeneServer class provides all the main functionality for JavaGene. It provides all necessary methods for the HTML user interface and to communicate with the outside resources.

A.2.1 User Interface Model

The basic user interface model for executing local scripts, GCG commands, and file operations is the same. One method displays a form gathering relevant information from the user. It then sends this input to a second method which handles it appropriately and displays the result to the user and updates the file display.

A.3 Utilities Class

The Utilities class mainly provides methods for manipulating strings. These methods simplify code in the other classes, and make it easier to read.

A.4 TextFile Class

The TextFile class extends the File class. It includes the ability to store the contents of a text file in a string, methods to read and write files to and from disk, and methods to set and get it's contents as well as the functionality provided by the File class.

A.5 gcgServer Class

The gcgServer Class provides one method, which can be invoked remotely using RMI. This method executes a requested GCG function on specified input files with a specified command line, and returns the expected output file to the calling method.

References

[1] European Bioinformatics Institute. *Sequence Retrieval System*. http://srs.ebi.ac.uk/.

[2] David Flanagan. *Java in a Nutshell*. O'Reilly and Associates, 2 edition, May 1997.

[3] Genetics Computer Group. *Genetics Computer Group*. http://www.gcg.com/.

[4] J. Kendrew, editor. *The Encyclopedia of Molecular Biology*. Blackwell, Oxford, 1994.

[5] S. E. Koonin. An independent perspective on the human genome project. *Science*, 279:36–37, 1998.

[6] Scott Trudeau. *JavaGene JavaDocs*. Edinburgh Parallel Computing Center. http://www.epcc.ed.ac.uk/ hwyau/JavaGene.

[7] University College London. *CINEMA Project*. http://www.biochem.ucl.ac.uk/bsm/dbbrowser/CINEMA2.1/.

[8] Clinton Wong. *Web Client Programming With Perl*. O'Reilly and Associates, 1 edition, March 1997.

Scott Trudeau is a student at Oakland University and Wayne State University in Detroit, Michigan, USA. He works at the Parallel and Distributed Computing Laboratory at Wayne State when he is not studying or overseas.

Supervised by Dr. Hon W. Yau of the Edinburgh Parallel Comupting Center and Dr. Martin Simmen of the Institute of Cell and Molecular Biology.

Photo here