

FunGIMS: A Functional Genomic Information Management System

A Meintjes, T de Beer, P Burger, C Hefer, A Oelofse, C Schriek,
A Myburg and F Joubert

Bioinformatics and Computational Biology Unit, University of Pretoria;
Department of Genetics, University of Pretoria, Pretoria 0002, South Africa

Introduction

While various Laboratory Information Management Systems exist, they are mostly of a commercial nature and deal with the management of data from automated machinery. This project deals with the development of a Functional Genomic Information Management System, which will attempt to integrate the management of data from a series of functional genomic-type studies, and eventually enable users to perform integrated queries on a systems level. It is based on a core module, which takes care of user authentication, group authentication, database interfacing, remote and local command execution and application interfacing. Upon this, a series of specialized modules are being constructed for dealing with different types of functional genomics data.

A primary consideration is to provide biologists with user friendly, integrated access to tools that they would otherwise probably not have attempted to use.

Overall Design

Some of the most important factors were that the system should be:

- Extensible (additional capability must be easy to add)
- Scalable (to large datasets, many users, etc.)
- Open (interface level must be documented and available)

The system is being developed in Java, using the Struts Open Source Framework. PostgreSQL is being used as the back-end database, with a Hibernate layer. Deployment will be in the Tomcat server environment (Figure 1).

Analysis Modules

The goal is not to re-design a series of new analysis tools, but to integrate various existing analysis tools in a local and remote fashion. The platform for the integration of these tools must be highly extensible, so that the addition of new tools should generally not involve modification of the source code. Analysis modules will include version-controlled annotation of results, complex linking between objects in the different modules, and automated re-running of analysis.

General Module

This module will store general methods, deal with storage of stocks, and deal with generic laboratory data such as gel images, chromatograms, and spreadsheet-type data.

Sequence Module

This module will store protein and nucleic acid sequence data, provide processing of trace files including basecalling, vector clipping, quality clipping and repeat masking. Passthroughs will be available for various standard types of analysis such as genefinding, promoter finding, BLAST searches, domain searches, motif searchers and alignment functions.

Structure Module

This module will provide access to structural information, structural display and manipulation features, mark-up of sequence features to structures and passthroughs to structural analysis, homology modeling, molecular dynamics and docking functionality.

Microarray Module

This module will provide storage of microarray data according to the MIAME standard, passthrough to analysis using the BioConductor tools, and mapping of results to GO terms and metabolic pathways.

Genotyping Module

This module will provide access to storage of genotyping results, and passthroughs to a series of RFLP and microsatellite analysis methods.

Other Modules

For future development, phylogenetics, proteomics and cheminformatics modules are proposed.

Discussion

Up to now, sections of the core module, and the data storage section of the microarray module have been implemented.

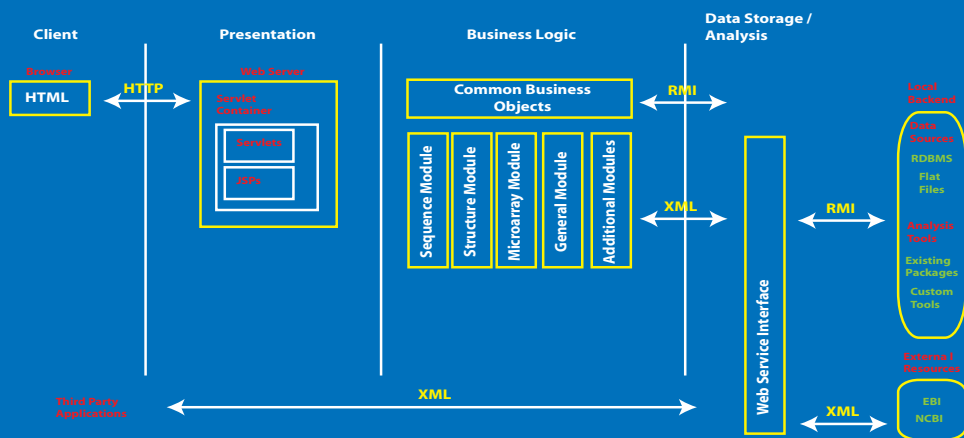


Figure 1: A diagrammatic representation of the architecture.



Bioinformatics and Computational
Biology Unit

Contact Information

Fourie Joubert
E-mail: fjoubert@postino.up.ac.za
<http://www.bi.up.ac.za>

