Christopher LoBianco

Advisor: Professor Christopher Armen; Professor Susan Bush

Seminar Instructor: Professor Ewa Syta

BioKit: An application for the integrated analysis of gene expression data

Midterm Progress Report

December 11th, 2018 (LATE)

Introduction

The purpose of this project is to build a desktop application that allows biological researchers to access and utilize computational biology tools irrespective of their level of technical (computer) training or experience. To be specific, this application will implement and encapsulate algorithms, packages, and tools for gene expression analysis. This process can be divided into four main stages: data processing, data analysis, database referencing, and visualization. Similarly, this application is also divided into four components that receive data input, execute the specified stage of gene expression data analysis, and pass output data to the next component (stage). Programmed in Java, this application takes advantage of numerous open source resources like BioConductor and BioJava. The final product will allows users to analyze gene expression data within the application while allowing the user to tailor the precise algorithms, statistical parameters, or visualization output to their particular experiment.

Implementation

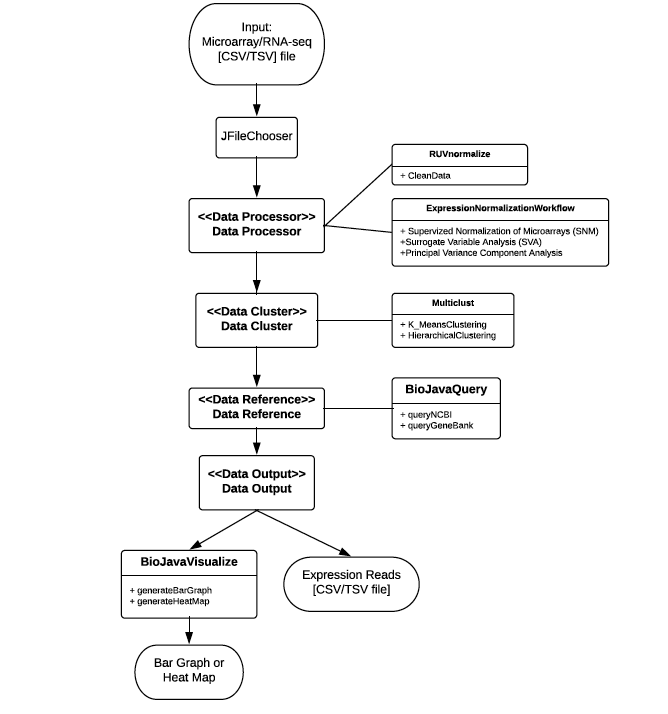
Implementation of this project can be best summarized as a work in progress. Although existing resources (BioConductor packages, BioJava API’s) are readily available and easy to access and implement, an effective application must identify the specific packages and modules that best solve the design challenge identified. For this reason, design and research has been the primary focus of the project up to this point. Having defined the project structure as one which parallels the four stages of gene expression data analysis, the most crucial stage of research and design was to identify, test, and evaluate the modules and packages necessary to implement each function. An overview of the project structure can be found on the followinTg page.

In terms of the modules that comprise this project, they can be grouped by the functions that they accomplish. Currently, the BioConductor packages necessary to analyze gene expression data are implemented and tested in R. In order to incorporate them into the application, further development of Java wrapper classes will be necessary. The modules that have been implemented and tested so far are as follows:

* **RUVnormalize**: tested on three different file input types (RNA-Seq CSV, Microarray CSV + TSV)
* **Expression Normalization Workflow**: tested two out of three statistical normalization techniques on sample data files
* **MultiClust**: read through documentation on two clustering methods used; researched statistical parameters most often used in experimental design/analysis
* **Other modules**: still evaluating BioConductor/BioJava packages from the currently available resources

In terms of challenges in implementing this software, the greatest challenge was choosing which module/package was best suited to the problem we wished to solve. There are literally thousands of possible BioConductor packages to choose from, and the selection process involved both communicating with biology faculty and an intense reading of biology research papers related to gene expression experiments. Additionally, not all packages use the same formatting for data input/output. This necessitates a careful review of modules, documentation, and the process of gene expression data analysis itself.

Figure : System Overview for BioKit



Preliminary Results

At this stage of the project, the underlying features have been implemented but have not been encapsulated into a functional, easy to use application for analyzing gene expression data. Multiple resources and modules exist for analyzing this data, and choosing the correct module is imperative for constructing an effective application. Current implemented (but not Java encapsulated) features are those related to data processing and analysis. The first module, DataProcessor, contains methods to identify and remove gaps and systemic variation from the inputted dataset. Also contained within this module are methods to normalize the data so that data files from different reads may be aggregated for cross-experimental analyses. The second module, DataCluster, contains methods for statistically clustering genes based upon similar expression patterns. This can be accomplished using two methods: K-means clustering and Hierarchical clustering. Both these methods are implemented by relevant BioConductor packages with the ability to specify statistical parameters prior to execution. The remaining modules are a work in progress, mainly because the BioConductor or BioJava packages that implement the desired features have yet to be identified or tested (for compatibility). A major challenge with this process is identifying whether or not tools are relevant, since the stated purpose (in the description) of a module is not always consistent with the documentation or the code. This necessitates a process of evaluation and testing before work on encapsulating and implementing (in Java) can be performed.

Conclusion and Future Work

Progress on this application can be best described as behind schedule. Researching relevant modules and packages took far longer than expected due to the variety of resources in BioConductor and a poorly defined experimental problem. After re-evaluating the purpose and functions of this application, this process became much easier, but this was done at the expense of time in the beginning stages of development. Successful completion of this project will demand increased effort as a result, and it is important that this effort be directed toward the completion of specific, deliverable features on designated dates. An updated timeline with precise definitions of deliverable features will follow in the very near future.