Due: August 23rd, 2014

Final Project Summary

Background

The project consists of providing a web utility for users to upload their RSEM assembly outputs for an initial visual examination of the reads before moving on to further processing, such as plotting or statistical analysis in R. The reason I chose this project stems from my recent exposure to the analysis of differentially expressed genes from microarray data, which incorporates a lot of data visualization properties and statistical analysis using R and Bioconductor. Even though R is a relatively easy language to learn, it still requires a learning curve that programmers with prior knowledge in procedural programming languages (e.g Java, Perl or Python) may not grasp rapidly.

A basic overview of the project consists of the user uploading two files (isoforms and genes), these files are parsed and values are entered into a custom MySQL database, where Transcripts ID are matched with the GeneID. The user is then taken to another page where (s)he can search transcript by ID, or can filter results based on gene's TPM or FPKM values. When results exceed 500 hits, only the top 500 hits are displayed on the browser to accommodate a reasonable web processing time. The resulting table will display the GeneID, TranscriptID, Gene_TPM, Transcript_TPM, Gene_FPKM, Transcript_FPKM, IsoPct. (Note: Transcript here refers to isoforms.) The columns can be expanded to include the length of each transcript, expected count, etc.

Use in the Field

Unlike microarray data, RNA-Seq outputs files are even larger (usually 10-20 million reads in FASTQ files) but the assembled reads from different assembly software are more reasonable to work with (e.g. TopHat, BowTie, RSEM). RSEM will estimate the gene and isoform expression levels from RNA-Seq data and outputs two main files: isoforms.results and genes.results. Instead of looking at these two files separately, they can be combined into one table for easier visualization and plotting. Alternatively, this can also be accomplished by loading these files into R, creating a new data matrix for downstream data analysis. However, this step can be time consuming since the matching of gene IDs with transcript IDs can be a non-intuitive task given the different size of each table. Relational database is used in the web utility as a storage table since it has linking and matching capabilities, making MySQL an attractive choice as a back-end utility. A custom schema is created and a diagram is found in the Appendix section.

Issues during implementation

One of the biggest issues during implementation is the 504 Gateway Timeout error I received on the browser while uploading my test files to the server. The process time from uploading the files, the subsequent parsing, and the loading of values into the database exceeded the time limit of the request. The testing of the scripts and loading the values into the database locally, the process took about 5 minutes, probably too long for the server. While the uploading of the original files into the server was successful, the parsing of the files followed by the connection to

the database, creation of tables, or clearing the tables values, and inserting new values were the delaying factors of the process. Since the 504 error is a server-side error, I cannot fix the problem unless I have administrative rights to the server. To bypass this issue, I truncated the RSEM outputs from an 18 mb file of ~200,000 reads into a workable file of less than 1mb containing ~10,000 reads.

Another issue I encountered is the directory permissions the Apache server can have access to. I had to find a world access directory separate from all my source code in order to save the uploaded files and write the output file to send to the user's email. Not doing so would result in errors that are not immediately apparent during execution in the web server.

Validation of Data with R and Conclusions

Since this is a fairly simple web tool, I loaded the transcripts into R for verification. The number of hits from the search by ID function and filter function match accurately. Overall I believe this can be a useful web utility if there were no web server constraints such as timeout limit, and restricted permissions.

Appendix

MySQL tables creation commands and diagram of schema.
The commands are also implemented in the upload_process_db.cgi script:

Database name: schang72

```
CREATE TABLE IF NOT EXISTS link (
        TranscriptID varchar(100) NOT NULL PRIMARY KEY,
        GeneID varchar(100) NOT NULL);
```

```
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| TranscriptID | varchar(100) | NO   | PRI | NULL    |       |
| GeneID       | varchar(100) | NO   |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
```

```
CREATE TABLE IF NOT EXISTS genes (
        GeneID varchar(100) NOT NULL PRIMARY KEY,
        Gene_length int NOT NULL,
        Gene_effective_length decimal(10,2) NOT NULL,
        Gene_expcount decimal(10,2) NOT NULL,
        Gene_TPM decimal(10,2) NOT NULL,
        Gene_FPKM decimal(10,2) NOT NULL);
```

```
+----------------------+--------------+------+-----+---------+-------+
| Field                | Type         | Null | Key | Default | Extra |
+----------------------+--------------+------+-----+---------+-------+
| GeneID               | varchar(100) | NO   | PRI | NULL    |       |
| Gene_length          | int(11)      | NO   |     | NULL    |       |
| Gene_effective_length | decimal(10,2) | NO   |     | NULL    |       |
| Gene_expcount        | decimal(10,2) | NO   |     | NULL    |       |
| Gene_TPM             | decimal(10,2) | NO   |     | NULL    |       |
| Gene_FPKM            | decimal(10,2) | NO   |     | NULL    |       |
+----------------------+--------------+------+-----+---------+-------+
```

```
CREATE TABLE IF NOT EXISTS isoforms (
        TranscriptID varchar(100) NOT NULL PRIMARY KEY,
        Trans_length int NOT NULL,
        Trans_effective_length decimal(10,2) NOT NULL,
        Trans_expcount decimal(10,2) NOT NULL,
        Trans_TPM decimal(10,2) NOT NULL,
        Trans_FPKM decimal(10,2) NOT NULL,
        IsoPct decimal(10,2) NOT NULL);
```

```
+-----------------------+--------------+------+-----+---------+-------+
| Field                 | Type         | Null | Key | Default | Extra |
+-----------------------+--------------+------+-----+---------+-------+
| TranscriptID          | varchar(100) | NO   | PRI | NULL    |       |
| Trans_length          | int(11)      | NO   |     | NULL    |       |
| Trans_effective_length | decimal(10,2) | NO   |     | NULL    |       |
| Trans_expcount        | decimal(10,2) | NO   |     | NULL    |       |
| Trans_TPM             | decimal(10,2) | NO   |     | NULL    |       |
| Trans_FPKM            | decimal(10,2) | NO   |     | NULL    |       |
| IsoPct                | decimal(10,2) | NO   |     | NULL    |       |
+-----------------------+--------------+------+-----+---------+-------+
```