A web-based tool for analysis of whole-genome DNA microarray datasets

Ben Kero

June 7, 2010
Contents

1 Introduction 3

2 Original Requirements Document 5
   2.1 Introduction ................................................. 6
   2.2 Project Description ...................................... 6
   2.3 Phase I Requirements ................................... 6
   2.4 Phase II Preliminary Requirements ..................... 8
   2.5 Phase III Requirements for Sharing Results .......... 8
   2.6 Sequence Diagram ......................................... 9
   2.7 Specific Tasks to be Undertaken ....................... 9
   2.8 Risk Assessment ......................................... 10
   2.9 Risk Response ........................................... 10
   2.10 Testing .................................................. 10
   2.11 Preliminary Timetable .................................. 11

3 Requirements Document Alterations 12

4 Weekly Reports 14
   4.1 Week 03 ................................................... 15
   4.2 Week 04 ................................................... 16
   4.3 Week 06 ................................................... 17
   4.4 Week 08 ................................................... 17
   4.5 Week 11 ................................................... 18
   4.6 Week 12 ................................................... 18
   4.7 Week 13 ................................................... 18
   4.8 Week 14 ................................................... 19
   4.9 Week 15 ................................................... 20
   4.10 Week 16 .................................................. 20
   4.11 Week 17 .................................................. 21
   4.12 Week 18 .................................................. 21
   4.13 Week 19 .................................................. 22
Chapter 1

Introduction
Brachy-interface is a tool requested by the Mockler Lab at Oregon State University’s Center for Genome Research and Biocomputing. It was requested so that scientists worldwide could use the CGRB’s computing cluster. Running large analysis jobs on large sets of genome data is a computationally expensive operation. A standard workstation could take days or weeks to complete a job that the compute cluster could execute in mere minutes.

An easy-to-use interface with a low barrier to entry was requested as the basic format for the project. With that minimal set of requirements it was decided that the main interface should be a web page capable of being displayed in all modern web browsers without the need for any extraneous plugins. The interface only requires a modern browser (Internet Explorer 5+, Firefox 1.5+, any version of Safari/Chrome/Opera). The interface was coded in HTML, CSS, and PHP.

This is important because not every researcher doing work on Brachypodium has the resources to afford a compute cluster in their own lab. The goal of the project is to allow these external researchers to be able to leverage the CGRB’s compute cluster, allowing external scientists to conduct research traditionally infeasible to them. This also helps the CGRB by gaining notoriety among the scientific community, as the first to implement such an interface for Brachypodium. Being first is a very important attribute, as many scientists will simply bookmark the interface and not seek out new (potentially improved) tools.

The mentor for this project was Todd Mockler, head of the Mockler lab, and Henry Priest, a member of the lab specializing in computer science. Additionally Larry Wilhelm and Douglas Bryant Jr were available for me to put questions to. Members of the project team included Ben Kero and Joe Chambers, however unforeseen circumstances caused Joe Chambers to abandon the project during the second stage. While present, roles of the team were such that Ben coded the interface and main parts of the interface’s pipeline execution. Joe was responsible for documentation and early interface mock-ups and design.

During the project the client played both an active role and supervisory role. Todd and Larry provided direction and guidance and eventually an interface pipeline for the BRAT analysis while Henry provided technical assistance whenever trouble with server permissions or other technical issues arose.
Chapter 2

Original Requirements Document
2.1 Introduction

In the past few years a grass plant called Brachypodium has emerged as a leading model species for basic scientific studies relating to the use of grasses as biofuel feedstocks. The Mockler laboratory (http://mocklerlab.cgrb.oregonstate.edu/) has played a leading role in sequencing and analyzing the Brachypodium genome and is working to develop tools for genomic analysis in this species. One of the tools being developed is called a DNA microarray. The DNA microarray technology involves representing all of the DNA in an organism (the genome) on a small (approximately 1cm²) glass surface. DNA microarrays have many uses in biological studies. For one example, they can be used to study when and where in an organism particular genes are turned on or off. We have developed a new microarray platform for Brachypodium. Because this is a novel tool, no bioinformatic tools yet exist for these microarrays.

2.2 Project Description

Ben Kero and I will be working with members of the Mockler laboratory at OSU (http://mocklerlab.cgrb.oregonstate.edu/) to develop a public web-based interface for automated statistical analysis of whole-genome DNA microarray datasets. We will be working with the premier model grass plant, Brachypodium distachyon which can be used by any scientist in the world who wants to analyze their DNA microarray data. Participating members of Mockler lab include Todd Mockler, Doug Bryant Jr., Larry Wilhelm, Harry Priest and Samuel Fox. In addition we will work with Christopher Sullivan, a Computational Scientist for the CGRB computing systems.

2.3 Phase I Requirements

A web-based interface that allows a user to upload and analyze Brachypodium distachyon whole-genome DNA microarray data control and experimental files, implemented on an OSU Center for Genome Research and Biocomputing (CGRB) server with computational processing provided by the CGRBs 400 node linux cluster and outputting results to the server and an email to the submitting customer.

1. When the user uses a web browser with Internet access, a web page providing an interface for uploading cel files for the analysis of Brachypodium distachyon whole genome DNA microarray datasets will be accessible to any user.
2. The interface will provide a location where the user can enter their email address.

3. The interface will provide a mechanism where the user can automatically create a condition space. The name of the condition space should be modifiable by the user.

4. The user should be able to browse their file system to select a file or group of files and display them as a collection in the user interface representing some condition.

5. The user should be able to label the selected files individually as some treatment label.

6. The user should be able to add additional files as part of the collection of files for a condition and be able to provide a treatment label for each additional file selected.

7. The user should be able to remove individual files from a condition which will also remove any corresponding labels.

8. The user should be able to remove an entire condition and its associated collection of files and treatment labels.

9. The user should be able to add as many conditions as they would like to the interface, but one at a time.

10. The user should be able to identify which of the defined conditions is the control condition, but there will only be one control condition for each set of experiments.

11. The user shall be able to initiate the transfer of identified files from the users file system to the CGRB file system.

12. The system should validate the format of the data files prior to being transferred.

13. The system should combine and compress the files before they are transferred.

14. The system should validate that the files transferred successfully and if not, should send a message to the user indicating such.
2.4 Phase II Preliminary Requirements

1. A method to select the reference (control) condition.

2. A method to select the optional treatment conditions that will be used for each experiment.

3. A method to select the various analysis tools to use for each experiment.

4. A method to select the parameters for each analysis tool to be used.

5. A method to initiate the analysis of the users data.

2.5 Phase III Requirements for Sharing Results

1. An email will need to be sent to the original user indicating that the analysis is complete.

2. Email should contain a link to the server location where the user can find their results.

3. Results may be in the form of JPEG, PNG or txt.
2.6 Sequence Diagram

![Sequence Diagram](image)

2.7 Specific Tasks to be Undertaken

1. Continue meeting with customer on a weekly basis.
2. Create use cases for the requirements.
3. Create Data Flow Sequence Diagram
4. Determine what software language will be used for the web application.
5. Develop Main Page Interface. (File upload)
6. Develop File Transfer Mechanism.
7. Determine how to identify which node to upload files to and which processor(s) to send the data to for analysis.
8. Develop Analysis Interface. (experiment Setup)
9. Develop Results Sharing interface
10. Prepare for Project Presentation.
2.8 Risk Assessment

1. Drops in the network when trying to transfer extremely large files.

2. Analyzer customers are not able to easily understand the system and upload the wrong information.

3. Files selected for transfer are formatted incorrectly.

2.9 Risk Response

1. We will be checking file sizes on both ends of the transfer to verify that the information has transferred completely. We will also compress the files before beginning the transfer.

2. We are performing usability testing on an ongoing basis to verify that the interface is understandable and easy to use. There will be documentation available to the end user to show them how to perform all tasks. If during this testing we see problems, we will know something is not happening the way it should and we can make corrections.

3. We will be verifying all files that are transferred to make sure they are formatted correctly before sending them on to the statistical analysis packages.

2.10 Testing

We will be performing usability testing with actual users who have been involved in the development of the product as well as general users who have no experience with our software. We will also be testing the transfer rates for files and the processing speed for the analysis. Each progressive piece of the application will be tested as it is developed as well as its interactions with existing pieces of the software.
Chapter 3

Requirements Document
Alterations
<table>
<thead>
<tr>
<th>Original Requirement</th>
<th>What happened</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine and compress files before transferred</td>
<td>Removed</td>
<td>This was unfeasible with current web standards. No browser implements this feature, and is unlikely to in the future. Keeping this requirement would mean relying on an obscure non-portable browser plugin.</td>
</tr>
<tr>
<td>Ability to label individual files</td>
<td>Altered</td>
<td>This feature was altered to being able to label individual file sets. Labeling individual files has no real value, as they are typically named after the file set, then enumerated.</td>
</tr>
<tr>
<td>Validate format of data before being sent</td>
<td>Removed</td>
<td>Same problem as compressing and combining. No browser API allows a web page to interact with files on it’s system without first uploading it.</td>
</tr>
</tbody>
</table>
Chapter 4
Weekly Reports
4.1 Week 03

Progress

On Monday October 10th, we contacted our customer to set up a meeting.

On Wednesday the 13th of October we met with Todd Mockler and the group which he represents from the OSU Center for Genome Research and Biocomputing. Included in the meeting were Todd Mockler, Doug Bryant, Larry Wilhelm, Henry Priest and Sam Fox. At that meeting we received a broad overview of the current state of array analysis for plants and the high level of fragmentation of data and services. They are looking to establish a more standardized environment where data and documentation can be accessed as well as analyzed.

In addition to standardizing the environment, we will be working on developing an analysis tool where users can upload raw data from customers via a web browser, start a number of standardized statistical analysis tools and send feedback to the user after the results become available.

We established Thursday @ 2:00pm as a current standard meeting time with Larry Wilhelm who is part of Todd’s group to discuss system issues and work on our requirements documentation.

On Thursday at 2:00 we met with Chris Sullivan to have our user accounts in the CGRB domain set up. Chris talked to us about the open nature of their work environment and what some of their requirements are related to how you use the system.

Ben and I are continuing to work on developing the system requirements and are starting to put together some general prototypes. Initially, we are just looking at access to the web site and a page for selecting files to be uploaded to the CGRB servers for analysis.

Problems

Plans

1. Develop requirements for the web site and analysis application. 2. Develop paper prototypes based on the requirements. 3. Develop the system specifications for using the site. 4. Develop the application specifications for using the analysis tool.
4.2 Week 04

Progress

We met with our customers again and both Ben and I have gained access to the needleman server as well the bishop server which is where we will doing our systems work. We have agreed to work using an agile development methodology and will be working on a week by week basis developing the requirements for the coming week. This week we will be creating a low fidelity prototype for the main page of the web application interface. From this page, the user will be able to browse their network and select the files they would like to upload to the CGRB. They will be able to select the files using a series approach. Created a prototype web page using VS2008 and posted that to the Google Group website. We also created an initial project management file using MS Project to track our progress and goals. Continuing to develop the customer requirements documentation and the project plan.

Problems

Plans

Start creating the use case diagrams for each piece of the application. Creating the low fidelity prototypes for the main interface page which contains the file selection and submission. Create the testing documentation for file selection and submission. Begin creating the requirements for the file transfer, establish checking methodology to confirm full transfer as well as checking to verify file formats. Progress Continuing to work on the MS Project spreadsheet. Adding more details as requested and including additional items for the May presentations. Met with some of our customers again to go over our initial web page suggestions for how the end user would select files for performing analysis. We generally received positive feedback with minor alterations, but the main customer who is the decision maker for this page was unable to be present. I have modified the original post with requested changes and we are awaiting feedback on this view of the main web page. Ben has set up a storage space in Beaversource where he and I can share files and conversations without including the customers. It also serves as an svn repository. Problems Plans We have been asked by Sorouch to work on the MS Project spreadsheet more as our main focus. Continuing to investigate whether we can zip files prior to transferring them to the OSU servers. We are continuing to address the main page prototypes as well and are loading our files to Beaversource as a common work location.
4.3  Week 06

Progress

We met with the clients again today. The client brought up interface mockups to the meeting. We brought some plausible code for their requirements. We went over our requirements and addressed a flow chart for the project. We met with our TA mentor and went over our gantt chart, and CRD.

Problems

Our TA told us to revise our gantt chart and make it less descriptive.

Plans

Revise our gantt chart, integrate the contents of the flow diagram into our CRD, insert all information into our central document repository (beaversource).

4.4  Week 08

Progress

Worked on the posterboard, 20% complete, Worked on the main web page, 20% complete. Created the use case diagrams for three use cases. Created Sequence diagrams. Approval of the experiment setup low fidelity prototype. Briefly discusses progress with customer team on Thursday.

Problems

Finding time to meet together and work on the project due to midterms and external meetings.

Plans

Scheduling working meetings with each other over weekend. Work on posterboard. Discuss the technology implementation plan Decide on format for experiment setup text file to be placed on server. Discuss script to read experiment setup file and execute analysis tools.
4.5  Week 11

Progress
Met with Ben Kero to discuss the team presentation for Tuesday 12 Dec 2010. Set up a doodle poll to establish new time for Array Analysis team meetings. No responses yet except for Ben and myself. Ben shared more files using svn. continuing to work on website coding. Rereading TRIP document.

Problems
no response from array analysis team members yet.

Plans
Create test documents for main page and file transfer. continue working on php and javascript for web pages.

4.6  Week 12

Progress
sent doodle poll to schedule array analysis team meeting. worked on php scripts for web pages continuing work on developing tests for web pages

Problems
problems getting responses from team members for doodle poll to schedule team meetings

Plans
continue work on testing for web pages and development of the pages as well as meeting with customers

4.7  Week 13

Progress
Met with customers to discuss the current status and to get additional information regarding the running of scripts on the compute cluster. They
indicated that Chris Sullivan had a computer fail and was working on get-
ing it up. They would try to escalate the issue of access to the cluster to run the scripts for analysis. Also indicated they would work on getting us files to use for testing the cel file transfer to the server.

Problems

Getting enough time to work on everything at once. Light load up till this point, and now everyone wants large results at the same time. Have to batten down the hatches. We had a sparse meeting with some of the customers. Received the information we needed, but Henry is working on getting us access to run the analysis packages and files so we can do some file transfer tests.

Plans

Need to work on the interface to get it completely up and running. Need to learn about JQuery library functions. We decided to use some JQuery in the web interface. Plan to meet with some of the resources this coming week.

4.8 Week 14

Progress

Received cel files for testing from Henry Priest. Still waiting on response from Chris Sullivan for access to run analysis packages. No meeting this past week due to other commitments.

Problems

Work load was pretty high this week and Joe had outside interviews for positions.

Plans

Test file transfer through interface for cel files provided by Henry and Sam. Test analysis package submission if we get access to run the packages.
4.9 Week 15

Progress
Ben and I exchanged ideas for the block diagram. Ben submitted the block diagram. Joe met with Larry Wilhelm to discuss the server side scripts that have to be executed from the web interface. He also had a short discussion with Todd Mockler. Ben uploaded his most recent changes to the web pages to Beaversource.

Problems
Ben and I need to work more closely and schedule time to work with each other. We need to make more progress on the web pages. We have files to use for testing and we need to get the web pages up and running so that we can test.

Plans
Get the main page up and running for condition definition and file selection. Get the submission page up and running for analysis tool selection and definition.

4.10 Week 16

Progress
Met with Joe and generated interface code. Added proper banner, used jquery to make dynamic forms and multiple file upload links. Ran all code through HTML validators and JSLint to ensure all code is valid.

Problems
Job submission has been less than straight forward. We will have to reverse-engineer some code in order to complete job submissions.

Plans
Meet with Joe again, and meet with clients to show them a finalized interface.
4.11  Week 17

Progress

Investigated file upload solutions for compression, convenience, and user interface. SVN repository cleaned up. Code tested on target web server.

Problems

Time conflicts affected scheduled meetings between Joe and Ben.

Plans

Implement file upload solution, focus on back-end processing. Test email notifications and SGE interface. Look into file conversions.

4.12  Week 18

Progress

Ben has been working on the web pages interface for submitting analysis to the OSU server as well as updating the Brachy poster. Joe has been working with Larry Wilhelm on getting the analysis running as a command line interface on the server and measuring times for transferring data files within various user environments. There is some sort of bug in running eset within R: eset <- rma(data) Todd Mockler has sent an email to Affymetrix asking for assistance.

Problems

Unable to get our web pages to display correctly within the university environment, from people.oregonstate.edu. Ben believes it is an issue with some configuration. There is some sort of bug in running eset within R: eset <- rma(data), Todd Mockler has sent an email to Affymetrix asking for assistance.

Plans

Address the problems with web browser display issues and get resolution on running eset. Continue to develop the web pages for test setup and
submission. Put together the rest of the post process interface on the server side and work on setting up the email for a completed submission.

4.13 Week 19

Progress

Worked on javascript to create dynamic interface. Ben and Joe collaborated to schedule a time to demo the project to Professor Bailey.

Problems

No physical meeting this week.

Plans

Demo the project to Professor Bailey. Continue coding.

4.14 Week 20

Progress

Ben met with our customer for a meeting that was scheduled at 2:00 on Friday to give them a demo and to go over changes they have requested in the requirements based on response they received from potential customers. I(Joe) had a class at that time so I went over and met with them afterwards. They would like us to step up the pace, but recognize that there is a certain portion of the initial request that cannot be implemented without changes that we are waiting on from Affymetrix. Todd has created a base level set of programs that he would like to go ahead and implement and Larry Wilhelm is working on a set of changes to their scripting programs that he can provide us to use in the implementation.

Problems

Delays as a result of 3rd party tool vendor needs. Affymetrix is redesigning the array cdf file.
Plans
Demo the project for Mike Bailey on Monday afternoon. Continue working on the web interface. Continue to work with Larry Wilhelm on the analysis scripts and with Henry Priest to roll out the web interface.

4.15 Week 21

Progress
Ben contacted Todd to give him a status update. Todd told him to coordinate with Henry to get our work integrated into their formal Brachypodium web site (brachypodium.org).

Problems
Member on Todd’s team never replied to the email so the work was never integrated.

Plans
Email Henry directly to schedule a meeting time for integrating the site.

4.16 Week 23

Progress
Improvements were made to the poster.

Problems
There was no communication between group members or clients.

Plans
Improve the poster, modify code to make it more robust or error resistant. Reattempt communication with clients.
4.17  **Week 24**

**Progress**
Met with clients, discussed future direction of the project.

**Problems**

**Plans**
Email the group for clarification on future work. Prepare poster.

4.18  **Week 25**

**Progress**
Contacted Henry, met new project member, Jess. Henry gave me the details on where to find his code and clarification about future work. Made finalized version of poster pending Professor Bailey’s review.

**Problems**
No problems encountered this week.

**Plans**
Meet with Professor Bailey to schedule a walkthrough of the project, and contribute improvements to the interface this week.

4.19  **Week 26**

**Progress**
Read Henry’s code. Started to write some improvements–some sources available at Beaversource. Prepared for expo showing.

**Problems**
Midterms this week. Bishop (the web server) is denying me access, so I’ve been looking over the externally viewable sources. needleman %7E 87% ssh
bishop Last login: Sun May 9 23:24:20 2010 from needleman.cgrb.oregonstate.edu
This account is currently not available. Connection to bishop closed.

**Plans**

Prepare more for expo showing. Wok on interface. Start a dialog with Chris about getting access on Bishop again.

**4.20 Week 28**

**Progress**

Coded modularity into the project to allow for extension of multiple tests. Documented code. Arranged for a meeting time to hand off continuation of project with Jess.

**Problems**

**Plans**

Finalize code, document more, and hand off code to Jess.
Chapter 5

Poster

Brachypodium DNA Microarray Analysis

In the past few years a grass plant called Brachypodium has emerged as a leading model species for basic scientific studies relating to the use of grasses as biofuel feedstocks. The Mockler laboratory is working to develop tools for genomic analysis in this species. One of the tools being developed is called a DNA microarray. The DNA microarray technology involves representing all of the DNA in an organism (the genome) on a small (approximately 1 cm) glass surface. The Mockler lab has developed a new DNA microarray platform for Brachypodium. This project will provide an important tool for the scientific community to analyze data produced on this DNA microarray platform.

Benjamin Kero: kerob@oregonstate.edu
Joe Chambers: chambjoe@oregonstate.edu
Mockler Team (Todd Mockler, Larry Willhelm, Henry Priest, Doug Bryant Jr., Samuel Fox)

Enabling scientists worldwide

A web interface was created using PHP, JavaScript and HTML scripting languages. The interface allows a user to upload data files, fine tune parameters, and submit compute jobs. The jobs are submitted to OSU’s CGRB server cluster, and processed. When the job is done, the system will generate a report. An obfuscated link is sent to the recipient, and is stored until retrieved.

Benjamin Kero
Joe Chambers
(not pictured)

Doing the Project

Interesting problems included which languages to code the tool in to have the most interoperability. Many problems were solved by PHP’s ability to “shell out” to other languages available on the system. We chose to do this because many analysis procedures used C, C++, Java, and Perl.

Results

We found that creating a web interface with the amount of dynamic content requested was non-trivial. Extensive programming knowledge and expertise were needed. The finished product contains six low-level utilities that are used for converting, filtering, trimming, and standardizing. Extensive programming knowledge was needed to interface with the GridSeerve, Linux, and bash. Extensive knowledge of programming languages was needed to develop the web interface.

Conclusions

* Output of a single test is often fed into other tests
* Other scientists are already using the tool
* Interfacing well with other software was paramount in achieving good results
* Given additional time compression of uploaded files would be the first stretch goal
* Goal of allowing novice analysis submissions to Sun GridEngine cluster was achieved

Interesting problems included which languages to code the tool in to have the most interoperability. Many problems were solved by PHP’s ability to “shell out” to other languages available on the system. We chose to do this because many analysis procedures used C, C++, Java, and Perl.
Chapter 6

Documentation
6.1 Project Overview

The brachypodium interface code is designed to run within the CGRB’s mason web server, although there is no reason why it cannot be run standalone, provided that script locations are altered (clearly marked at the top of each file/function). Environmental requirements for the software include Sun GridEngine and a web server capable of handling PHP.

6.2 Installation

1. Ensure that a subversion client is installed on your computer.

2. Retrieve the Brachy project subversion repository from BeaverSource repository using the "svn co –username https://code.oregonstate.edu/svn/brachy” command from a Unix terminal. If checking out on Windows consider using a integrated Subversion client such as TortoiseSVN(http://tortoisesvn.tigris.org/).

3. Move the contents of the directory somewhere to a web root (such as $HOME/public_html/brachy).

4. Access the installed files by visiting the web-accessible location in your web browser (such as http://onid.oregonstate.edu/username/brachy).

5. If using external to the CGRB’s compute cluster, change the test script locations at the top of brat.php, and process.php.

6. Please note that the default PHP max upload file size is 2 Megabytes. It is recommended to boost max_upload_filesize in /etc/php.ini to a value of around 100M.

6.3 Usage

Access the interface by opening up the index page in your web browser. To do this you’ll need to remember the location that you installed the files. If you followed the example from the installation directions above, it should be located at http://onid.oregonstate.edu/username/brachy/index.html.

From here you will be presented with the main interface. Notice the single section for uploading a control file set. The default section only has a slot for one file. Additional files can be added by clicking the "Add File" link in the section. To add another file set to add to the job, click the "Add Group" link directly blow the file sets.
To submit a job, add at least 1 file set, select files to upload, and give each group a name. Ensure that the Control radio box is selected in the right set. Notice the "Config" tab located at the top of each file set. In future versions, this will be able to specify options associated with each file set. Below the file sets, fill in your e-mail address, an optional job name, and a normalization method. Click the 'Submit' button at the bottom of the screen, and your job will be submitted. If no error was given, an email should be sent to the e-mail address specified when the job is submitted, and an additional time when the job is completed. Take note of the job name, and refer to it if you are requesting the status of a job.

To request the status of a submitted job, click the "Get Status" link at the top of the interface. This will present you with a short paragraph of instruction to enter the job name and click the 'Submit' button. The status for the submitted job should be retrieved. Please note that completed job results stay cached for up to 2 weeks. After 2 weeks please submit the job again.

6.4 File Descriptions

- index.html - main web page
- index.js - logic to make index.html dynamic
- index.css - Make index.html pretty
- tabber.js - javascript library to provide tab functionality
- jquery.js - javascript library to provide dynamic functionality
- lib.php - provide standard library functions to the other PHP scripts
- process.php - PHP script to submit jobs requests and handle uploaded files.
- brat.php - Handles the BRAT upload pipeline.
- status.php - Get a job’s status or link to a completed job’s results
- get_output.php - Retrieve a completed job’s results
Chapter 7

Technology
For development of the project many sites were used, some of which were very helpful providing references and example code, and others that were not nearly as helpful. One of the most comprehensive and often used resources for this project was http://www.php.net. PHP.net is the official site of the PHP scripting language, and includes very helpful features, such as a complete function listing available as directories in the URI, such as the documentation listing for fopen(), a function used to open files. The fopen() documentation listing is available at http://www.php.net/fopen. Other functions are reachable using the same method.

http://www.jquery.com was also used extensively for development of the interface. Many of the dynamic functions of the page were only possible with the JQuery library. Henry Priest, a member from the Mockler Lab was invaluable for overcoming some of the issues that came up during development, such as fixing permission issues for web roots when uploading test scripts.

What follows is a listing of the reference and tutorial sites used for the project:

- http://www.php.net/
- http://www.jquery.com/
- http://www.w3schools.com/PHP
- http://www.w3schools.com/JavaScript
- http://www.w3schools.com/CSS
- http://genome.cgrb.oregonstate.edu/using.html
Chapter 8

Learned
There was much technical information to learn in order to complete this project. Entering the project, either project member had any experience in the realm of biocomputing or genomics research before, and were completely unfamiliar with the tools. Additionally, neither member had used the Mason web framework before. The Sun GridEngine clustering software was also new, and required learning before being able to use it.

The project also taught the group members about working with deadlines. Deadlines were used to stage different sections of the project, and to separate it into workable chunks. At each deadline was a set of deliverables that needed to be handed in. The weekly P3’s were used as a structured journal to reflect on development and used to view the project from an external third-person viewpoint.

Things learned in relation to the project work involve the aforementioned deadlines, deliverables, and technical skill involved. Although the project looked daunting when viewed as a whole, once broken down into pieces, it was possible to complete them separately, and then tie them together at the end.

Project management played an important part in this project as well. Both team members and the client played parts as managers. The team members managed each other when one seemed not to be producing deliverables in time, while the clients managed by assigning deliverables and giving direction and clarification.

Working in teams was a new experience for both team members. The team had a unique opportunity to reflect upon working in a team as opposed to working alone because one of the team members abandoned the project half-way through. In some ways, working in a group is more effort because one member must be sure not to "step on the toes" of the other member. Additionally, documentation must be more comprehensive because one member of the group might not be familiar with a part of the program created by another member. This was the case with the project, as Joe was unable to read and interpret the code until thorough documentation was provided.

If given another chance to complete the project, the group should have communicated more with the clients, gotten a more detailed outline of the work to be done, and divided the work into more serviceable chunks. More communication needed to happen over e-mail, and correspondence should have been faster in some cases.
Chapter 9

Essential Code Listings
9.1 process.php

<?php

include "lib.php";
include "brat.php";
include "page.php";
include "limma.php";
include "sam.php";

print_r($_POST);
print_r($_FILES);
if (!isset($_POST['email']) || !isset($_POST['brat'])
    || !isset($_FILES['group0upload']) || !isset($_FILES['group1upload'])) {
    echo "Error: Parameter missing. Ensure that a minimum of 2 sets of files are uploaded, at least one analysis is checked, and an e-mail address is supplied.\n"
    die;
}

isset($_POST['jobname']) ? $jobid = $_POST['jobname'] : 
    $jobid = rand_str(8);

$email = $_POST['email'];
$jobdir = "/local/cluster/www_rslt/$email-$jobid";
mkdir($jobdir) or die("Cannot create directory $jobdir");

# Test the files for size and contents
if (test_files($_FILES) == false) {
    cleanup_folder($_FILES) or die("Unable to clean up files!");
}
rmdir($jobdir) or die("Unable to remove directory " . $jobdir"; 

die("A file failed testing. Job aborted.

{"}

# Files check out, move them into the job directory
foreach ($FILES as $files) {
   for ($i=0; $i < count($files['name']); $i++)
      move_uploaded_file($files['tmp_name'][$i], $jobdir."/$files['name'][$i]) or die("Unable to move file " . $files['tmp_name'][$i]);
}

# Test Definitions
if (isset($POST['brat'])) {
   generate_brat_config($POST, $FILES, $jobdir) or 
      die("Failed to generate BRAT config file. Job aborted.

brat_run($jobdir, $POST['normalization'], $email, $jobid) or die("Failed to start BRAT job on the cluster.

if (isset($POST['page'])) {
}

if (isset($POST['limma'])) {
}

if (isset($POST['sam'])) {

} 

send_mail($email, $jobid);
?>

<body>
Your job was successfully submitted. Job ID is `<?php echo $jobid ?>`. An email confirming your submission was sent. Another email will be sent to you when the job is completed.

Contents of the config file are as follows:

```php
<?php
    $fh = fopen("$jobdir/config.txt", "r");
    print fread($fh, filesize("$jobdir/config.txt"));
?>
```

9.2 brat.php

```php
function brat_run($directory, $normalization_method, $email, $jobid) {
    $qsub_script = "/raid1/home/cgrb/cgrblib-dev/perl5/Diurnal perl5/pipeline-bin/bd_qsub.pl";
    $mailfrom = "brachypodium.org";
    $mailprog = "/usr/sbin/sendmail";

    $script = "/raid1/home/cgrb/cgrblib-dev/perl5/Diurnal perl5/pipeline-bin/bd_pipeline.pl";
    $script += ":-f_config.txt:-d$_directory";

    if ($normalization_method == "median") {
        $script += ":-m";
    }
```
```php
$fh = fopen($directory . "/bdarray_qsub.cluster");
fwrite($fh, "#!/bin/csh
" );
fwrite($fh, "#!/bin/sh
" );
fwrite($fh, "#!/N BDARRAY
" );
fwrite($fh, "#!/o $directory
" );
fwrite($fh, "#!/e $directory
" );
fwrite($fh, "#!/p 10
" );
fwrite($fh, " $script
" );
fwrite($fh, " echo 'Your job with job ID $jobid is completed. Please obtain your results from http://arrays.brachypodium.org:8080/testing/brachy/get_status.php?email=$email&jobid=$jobid' | mail-S $email

# qcmd will run $script
$qcmd = "$qsub_script $directory/$email $mailfrom $mailprog";
if (exec("perl $qcmd")) {
    return true;
} else {
    return false;
}
}

function brat_generate_config($postdata, $files, $directory) {
    $configfile = fopen($directory . "/config.txt", 'w')
or print ("Can't open config file: $directory/config.txt");

    if (!isset($configfile)) {
        return false;
    }
}

# Write the header
fwrite($configfile, "Name\tFileName\tTarget\n" );

# Identify the control group
$control = $postdata["control"];

# Add the control group
$controlgroupname = "group" . $control . ".upload";
```
# Set the repetition counter to 0
$repetitioncounter = 0;

for ($i=0; $i < count($files[$controlgroupname]["name"]); $i++) {
    # $line = $postData["title".$control.$$i."\t".$files[$controlgroupname]["name"][].$$i."\t".$cr$$i\n ";
    $line = "Control\t".$files[$controlgroupname]["name"][].$$i."\t".$cr$$i\n";
    fwrite($configfile, $line);
    $repetitioncounter++;
}

# We don't need the control group anymore, it'll just get in our way!
unset($files[$controlgroupname]);

# Now we add each file in the remaining groups
$i = 1;
foreach ($files as $groupname => $filesset) {
    $title = "title".$groupname[5];
    for ($j=0; $j < count($filesset["name"]); $j++) {
        $line = 
         "$postData[$title]$$j\t".$files[$groupname]["name"][].$$j."\t".$tc.$$i."r".$$j."\n";
        fwrite($configfile, $line);
    }
    $i++;
}

fclose($configfile);

return true;

?>