# Online Bioinformatics Toolkit

CYCLE 2 REPORT

Group 9

**Mason Monday, Sadaira Packer, Paul Ryu, Ben Williams, Ansleigh Yancey**

## image_guitar.jpg

**Submitted in partial fulfillment of the requirements for COMP 4710 Senior Design to the Department of Computer Science and Software Engineering, Samuel Ginn College of Engineering, Auburn University**

**Auburn, Alabama**

**April 1, 2019**

TABLE OF CONTENTS

horizontal line

1. System Metaphor...............................……………………………………………………………………...…. 3
2. Cycle Intent …………………….……………………………………………………………………………………..….… 3
3. User Stories…………………...…………………………………..………………………………………………….. 4-13
4. Design Documentation....…………………………...……………………..…………………………….…… 14-16

4.1 Architecture

4.2 Structure

4.3 Data Manipulation and Storage

1. User Interface....………………...……………………...……………………..…………………………….…… 17-19

5.1 Visuals

5.2 User Requirements

1. Management Plan………………...…………………………………………………………………………..…. 20-21
   1. Task Assignments
   2. Development Schedule
   3. Planned Code / Feature Freeze
2. Testing Plan …………………………..…………………………………………………………………………..…. 22-23
3. Risk Mitigation…………………………..…………………………………………………………………………..….. 24
4. Lessons Learned …………...…………………………………………………………………………………………… 25

1. SYSTEM METAPHOR

horizontal line

There is a lack of methods that organize genetic variants for diseases, tissues, and cell types. Advanced computational methods and software are needed to create a database and online web server that can address these issues. Our objective is to develop an interactive and online database that is searchable by disease type and/or tissue and cell line. Users will also be able to request the entry or removal of information in the database.

*By: Sadaira Packer*

2. CYCLE INTENT

horizontal line

Our main goal for Cycle 2 was to complete the implementation of search functionality based on cell line and/or disease type by intersecting two relevant datasets. We also intended to develop a process by which a user could dynamically replace one of the two datasets with a custom dataset.

*By: Benjamin Williams*

3. USER STORIES

horizontal line

|  |  |
| --- | --- |
| **User Story #1** | Add Entry Request (Text Box) |
| **Description** | The actor sends a request to add an entry to the database using the text box. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor chooses the “send add request” option. 3. The actor enters information on a genetic mutation or genome signal into the provided text box. 4. The actor’s request to add the entry to the database is sent to the administrator. |
| **Extensions** | 1a. Failure to send entry request:   * The text box entry was empty, and no request was sent to the administrator. |
| **Post-Conditions** | 1a. The Entry is added to the database:   * The administrator reviewed the actor’s request and permitted its addition to the existing database. * The requesting actor is notified.   1b. The Entry is not added to the database:   * The administrator reviewed the actor’s request and did not permit its addition to the existing database. * The requesting actor is notified. |
| **Hours** | Total Planned Hours: 30  Planned This Cycle: 5  Total Actual: 6.5  Actual This Cycle: 0 |
| **Coder** | Ansleigh Yancey (Cycle 1) |
| **Tester** | Ansleigh Yancey (Cycle 1) |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #2** | Add Entry Request (Text File) |
| **Description** | The actor sends a request to add an entry to the database by uploading a .txt file. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor chooses the “send add request” option. 3. The actor uploads a file with information on a genetic mutation or genome signal. 4. The actor’s request to add the entry to the database is sent to the administrator. |
| **Extensions** | 1a. Failure to send entry request:   * No file was uploaded and no request was sent to the administrator. * The file uploaded was not in .txt format. |
| **Post-Conditions** | 1a. The Entry is added to the database:   * The administrator reviewed the actor’s request and permitted its addition to the existing database. * The requesting actor is notified.   1b. The Entry is not added to the database:   * The administrator reviewed the actor’s request and did not permit its addition to the existing database. * The requesting actor is notified. |
| **Hours** | Total Planned Hours: 30  Planned This Cycle: 5  Total Actual: 4  Actual This Cycle: 0 |
| **Coder** | Ansleigh Yancey (Cycle 1) |
| **Tester** | Ansleigh Yancey (Cycle 1) |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #3** | Add Entry Via Text Box (Administrator) |
| **Description** | The actor adds an entry to the database using the text box. |
| **Actor** | Administrator |
| **Precondition** | The actor is connected to the internet and has navigated to the website and is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the admin webpage. 2. The actor chooses the “add entry” option. 3. The actor enters information on a genetic mutation or genome signal into the provided text box. |
| **Extensions** | 1a. Failure to add entry:   * The text box entry was empty, and no entry was added. |
| **Post-Conditions** | The Entry is added to the database |
| **Hours** | Total Planned Hours: 20  Planned This Cycle: 5  Total Actual: 5  Actual This Cycle: 0 |
| **Coder** | Ben Williams (Cycle 1) |
| **Tester** | Ben Williams (Cycle 1) |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #4** | Add Entry Via File (Administrator) |
| **Description** | The actor adds an entry to the database by uploading a file. |
| **Actor** | Administrator |
| **Precondition** | 1a. The actor is connected to the internet and has navigated to the website.  1b. The actor is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the admin webpage. 2. The actor chooses the “add entry” option. 3. The actor uploads a file with the information on a genetic mutation or genome signal. |
| **Extensions** | 1a. Failure to add entry:   * The file uploaded was not in the .txt format, and no entry was added. |
| **Post-Conditions** | The Entry is added to the database |
| **Hours** | Total Planned Hours: 20  Planned This Cycle: 5  Total Actual: 0  Actual This Cycle: 0 |
| **Coder** | N/A |
| **Tester** | N/A |
| **Reviewer** | N/A |

|  |  |
| --- | --- |
| **User Story #5** | Delete Entry Request |
| **Description** | The actor sends a request to delete an existing entry from the database. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor finds the entry that he/she would like to be removed. 3. The actor chooses the “send delete request” option. 4. The actor enters his/her reasoning for deletion (optional). 5. The actor’s request to delete the existing entry is sent to the administrator. |
| **Extensions** |  |
| **Post-Conditions** | 1a. The Entry is deleted from the database:   * The administrator reviewed the actor’s request and permitted its removal from the database. * The requesting actor is notified   1b. The Entry is not deleted from the database:   * The administrator reviewed the actor’s request and did not permit its removal from the database. * The requesting actor is notified. |
| **Hours** | Total Planned Hours: 30  Planned This Cycle: 5  Total Actual: 5  Actual This Cycle: 0 |
| **Coder** | Ansleigh Yancey (Cycle 1) |
| **Tester** | Ansleigh Yancey (Cycle 1) |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #6** | Delete Entry (Administrator) |
| **Description** | The actor deletes an existing entry from the database. |
| **Actor** | Administrator |
| **Precondition** | 1a. The actor is connected to the internet and has navigated to the website.  1b. The actor is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor finds the entry that he/she would like to be removed. 3. The actor chooses the “delete entry” option. |
| **Extensions** |  |
| **Post-Conditions** | 1a. The Entry is deleted from the database:   * The administrator removed the desired entry from the database. |
| **Hours** | Total Planned Hours: 20  Planned This Cycle: 5  Total Actual: 5  Actual This Cycle: 0 |
| **Coder** | Ben Williams (Cycle 1) |
| **Tester** | Ben Williams (Cycle 1) |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #7** | Edit Entry Request |
| **Description** | The actor sends a request to edit an existing entry from the database. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor finds the entry that he/she would like to edit. 3. The actor chooses the “send edit request” option. 4. The actor enters his/her requested changes and his/her reasoning for said changes in the provided text box. 5. The actor’s request for the existing entry to be edited is sent to the administrator. |
| **Extensions** | 1a. Failure to add entry:   * The text box entry was empty, and no edit request was sent. |
| **Post-Conditions** | 1a. The Existing Entry is Edited:   * The administrator reviewed the actor’s request and permitted the suggested changes. The entry is edited on the database. * The requesting actor is notified.   1b. The Existing Entry is Not Edited:   * The administrator reviewed the actor’s request and denied the suggested changes. The entry remains the same on the database. * The requesting actor is notified. |
| **Hours** | Total Planned Hours: 30  Planned This Cycle: 5  Total Actual: 0  Actual This Cycle: 0 |
| **Coder** | N/A |
| **Tester** | N/A |
| **Reviewer** | N/A |

|  |  |
| --- | --- |
| **User Story #8** | Edit Entry (Administrator) |
| **Description** | The actor edits an existing database entry. |
| **Actor** | Administrator |
| **Precondition** | 1a. The actor is connected to the internet and has navigated to the website.  1b. The actor is an administrator of the site. |
| **Basic Flow of Events** | 1. The actor navigates to the admin webpage. 2. The actor finds the entry he/she desires to edit. 3. The actor chooses the “edit entry” option. 4. The actor alters the existing entry’s information using the text box. |
| **Extensions** | 1a. Failure to add entry:   * The text box entry was empty, and no edit was submitted. |
| **Post-Conditions** | 1. The Entry is edited:  * The Administrator’s changes to the existing database entry are saved. |
| **Hours** | Total Planned Hours: 25  Planned This Cycle: 5  Total Actual: 0  Actual This Cycle: 0 |
| **Coder** | Ben Williams (Cycle 1) |
| **Tester** | Ben Williams (Cycle 1) |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #9** | Search for Database Entry (Cell Line) |
| **Description** | The actor searches the database for the desired genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “cell line” drop down box. 3. The actor chooses the “search database” option. |
| **Extensions** | 1a. Failure to search for entry:   * The search options were incorrectly completed. * The search options were incomplete. |
| **Post-Conditions** | 1a. The Existing Entry is Found:   * The search options for the cell line yields one or more results. * The information is displayed for the actor.   1b. The Existing Entry is Not Found:   * The search options chosen by the actor do not yield a result. * There is no existing entry in the database. |
| **Hours** | Total Planned Hours: 40  Planned This Cycle: 15  Total Actual: 59.5  Actual This Cycle: 12.5 |
| **Coder** | Ansleigh Yancey (Cycle 1)  Team (Cycle 2) |
| **Tester** | Mason Monday |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #10** | Search for Database Entry (disease type) |
| **Description** | The actor searches the database for the desired genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “disease type” drop down box. 3. The actor chooses the “search database” option. |
| **Extensions** | 1a. Failure to search for entry:   * The search options were incorrectly completed. * The search options were incomplete. |
| **Post-Conditions** | 1a. The Existing Entry is Found:   * The search options for the disease type yields one or more results. * The information is displayed for the actor.   1b. The Existing Entry is Not Found:   * The search options chosen by the actor do not yield a result. * There is no existing entry in the database. |
| **Hours** | Total Planned Hours: 40  Planned This Cycle: 15  Total Actual: 63.5  Actual This Cycle: 16.5 |
| **Coder** | Ansleigh Yancey (Cycle 1)  Team (Cycle 2) |
| **Tester** | Mason Monday |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #11** | Search for Database Entry (cell line & disease type) |
| **Description** | The actor searches the database for the desired genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “cell line” and “disease type” drop down boxes. 3. The actor chooses the “search database” option. |
| **Extensions** | 1a. Failure to search for entry:   * The search options were incorrectly completed. * The search options were incomplete. |
| **Post-Conditions** | 1a. The Existing Entry is Found:   * The search options for the cell line and disease type yields one or more results. * The information is displayed for the actor.   1b. The Existing Entry is Not Found:   * The search options for the cell line and disease type do not yield a result. * There is no existing entry in the database. |
| **Hours** | Total Planned Hours: 40  Planned This Cycle: 15  Total Actual: 66.5  Actual This Cycle: 22.5 |
| **Coder** | Ansleigh Yancey (Cycle 1)  Team (Cycle 2) |
| **Tester** | Mason Monday |
| **Reviewer** | Team |

|  |  |
| --- | --- |
| **User Story #12** | Download Search Results |
| **Description** | The actor downloads information about a genetic mutation and/or human genome signal. |
| **Actor** | User |
| **Precondition** | The user is connected to the internet and has navigated to the website. |
| **Basic Flow of Events** | 1. The actor navigates to the webpage. 2. The actor completes the search entry based on predefined search options in the “cell line” and/or “disease type” drop down boxes. 3. The actor chooses the “search database” option. 4. The actor chooses the download type (excel, text, word document). |
| **Extensions** |  |
| **Post-Conditions** | 1. The Existing Entry is Found:   * The actor saves the file to his/her personal device. |
| **Hours** | Total Planned Hours: 40  Planned This Cycle: 5  Total Actual: 2  Actual This Cycle: 0 |
| **Coder** | Ansleigh Yancey (Cycle 1) |
| **Tester** | Team |
| **Reviewer** | Team |

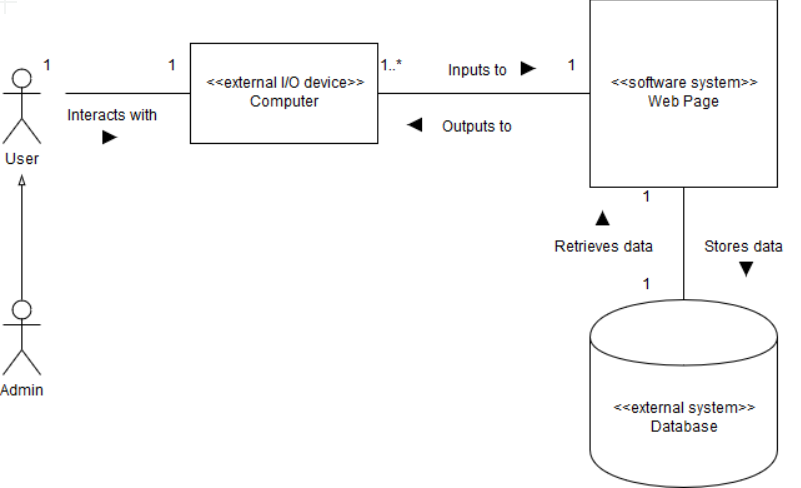
*By: Ansleigh Yancey*

4. DESIGN DOCUMENTATION

horizontal line

**4.1 Architecture:**

The human genome data is stored on a database developed in R. This database is linked to a webpage developed with R and hosted on the web server Shiny that our users will interact with. Users will input the values for the specific piece of data they wish to access into the webpage, which will then search our database for the data. Once found, the webpage retrieves the data and displays it to the user. Also, users may request changes to the database, such as adding a new entry. Upon submitting a change, the webpage holds the change until it is approved, after which the webpage will edit the desired entry.



This system context class diagram represents purely the external interfaces by which the user interacts with the system.

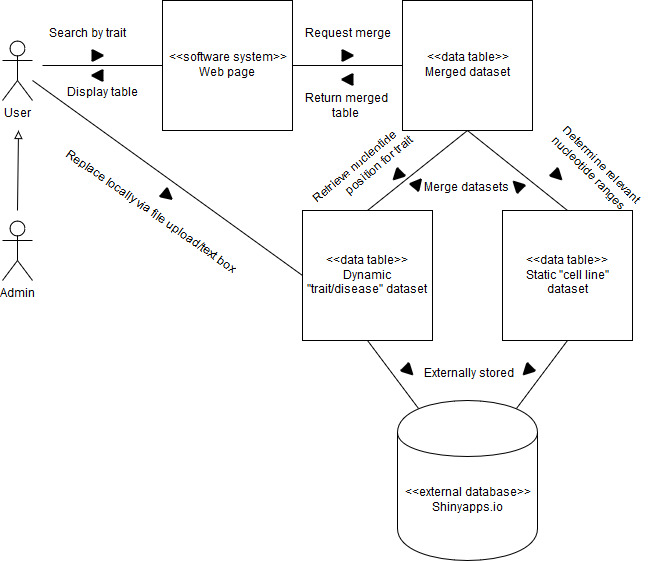
**4.2 Structure:**

The database application will be created using the R Shiny framework, and will be hosted using shinyapps.io. The primary, static dataset will be contained within a table containing seven columns and a row for each entry, constructed with the following attributes:

* ID: the unique value assigned to each entry in the database
* chrom: the name of the chromosome or scaffold
* chromStart: the starting position of the chromosome or scaffold
* chromEnd: the ending position of the chromosome or scaffold
* name: the name of the genetic variant
* score: the score given to the genetic variant
* blocks: the specific block the genetic variant resides on

Users may submit requests to add and delete entries. Upon approval by an administrative party, the table will be updated. The majority of the UI and server, as well as methods to CRUD (create, read, update, or destroy entries) will be hosted in a file called “main.R” . The table will, by default, be populated with just over 4.3 million entries from a provided .bed file. As development will be done in the RStudio IDE using Git as the primary file sharing service, the system will also contain default RStudio project files and a Git repository. Documentation on the table entries can be found at <https://genome.ucsc.edu/FAQ/FAQformat.html#format1>.

**4.3 Data Manipulation and Storage:**

****

This diagram represents how data will be stored and manipulated upon user request. The dynamic dataset will have a default copy hosted on Shinyapps.io, but a user may locally replace their dataset via a file upload or a text box. When a user requests to search by a specific trait, the request will propagate through the system, eventually merging the static and dynamic datasets based on the relative positions of nucleotides on a chromosome.

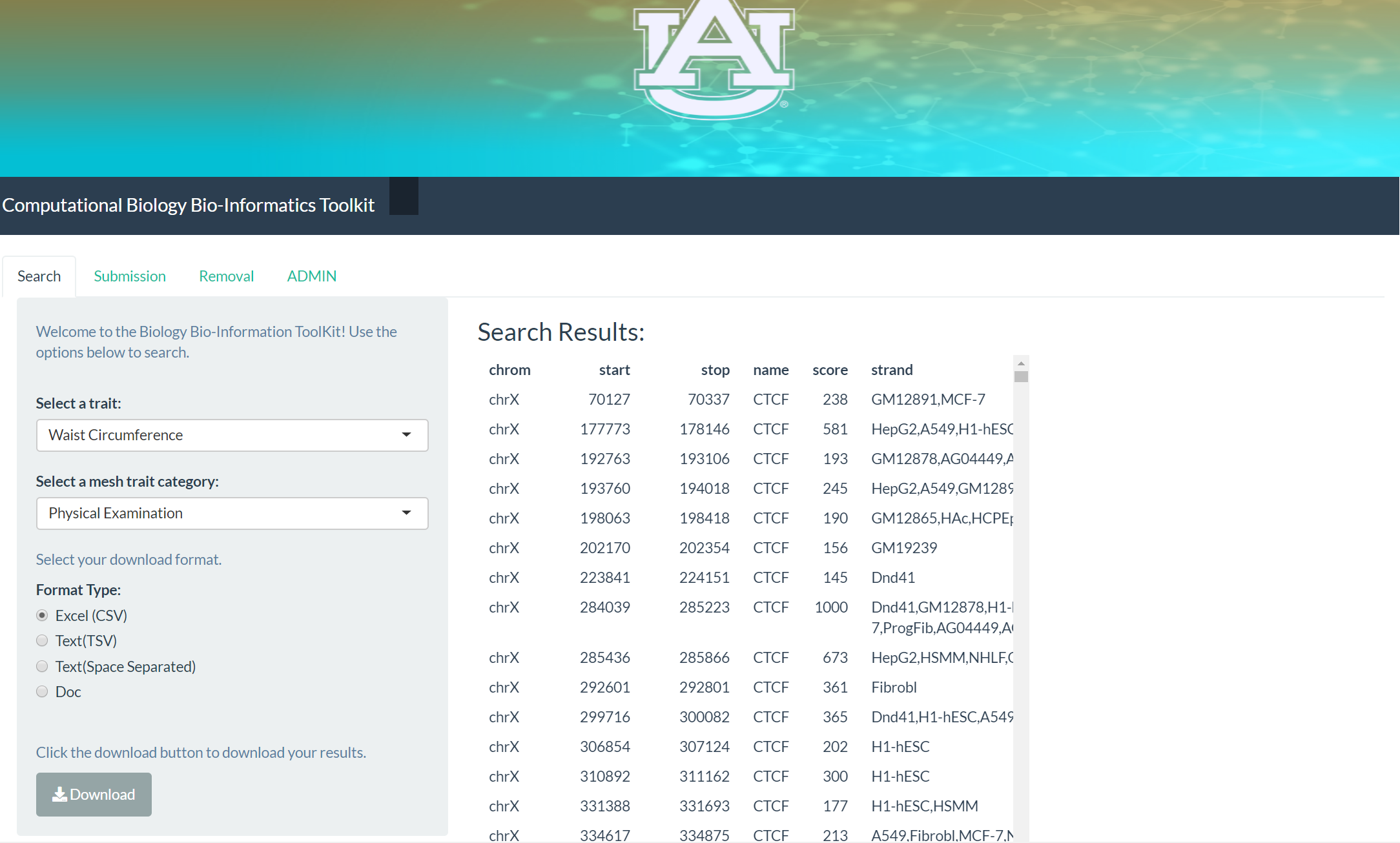
*By: Benjamin Williams*

5. USER INTERFACE DESIGN

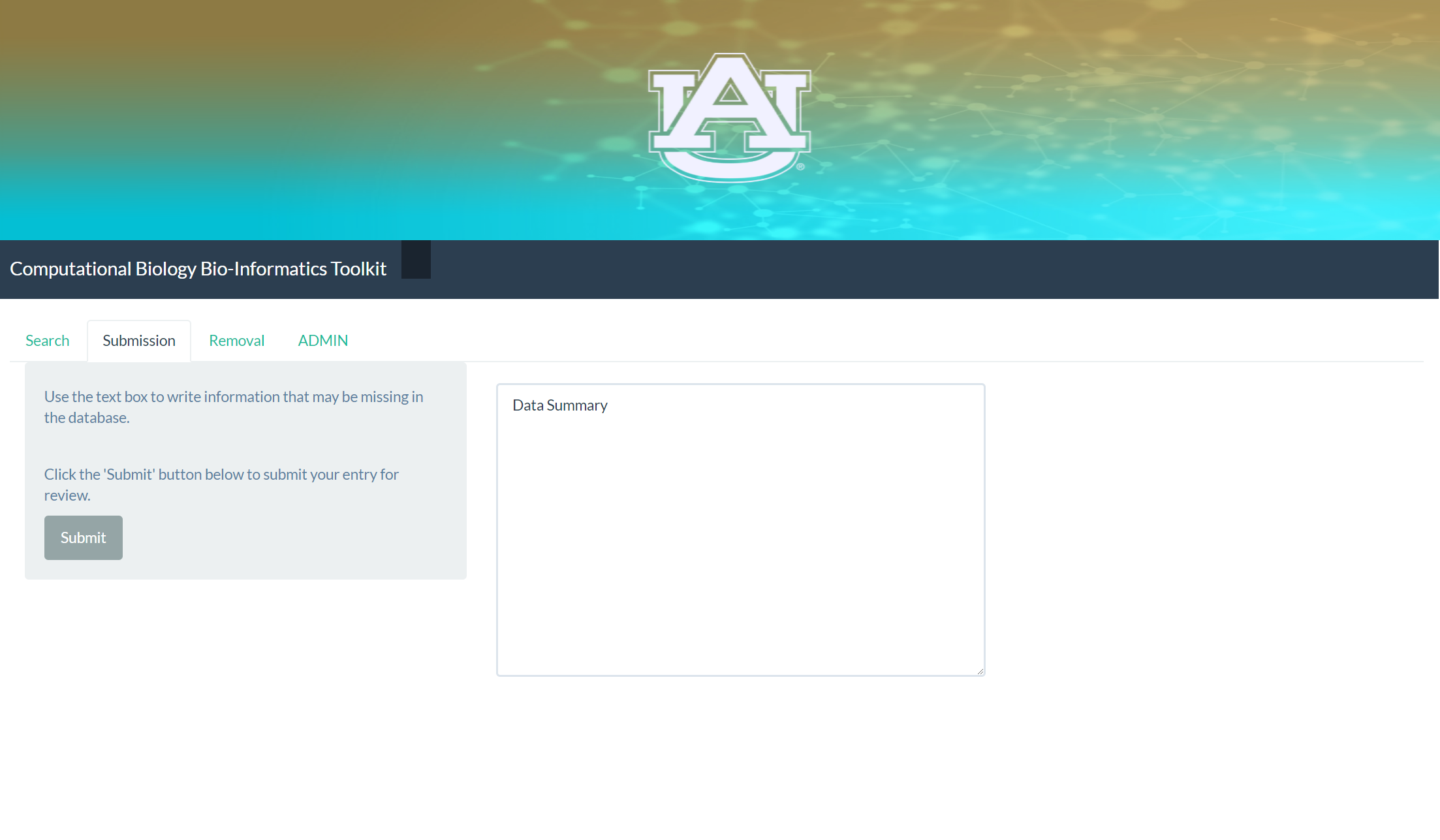
horizontal line

**5.1 Interface Visuals:**

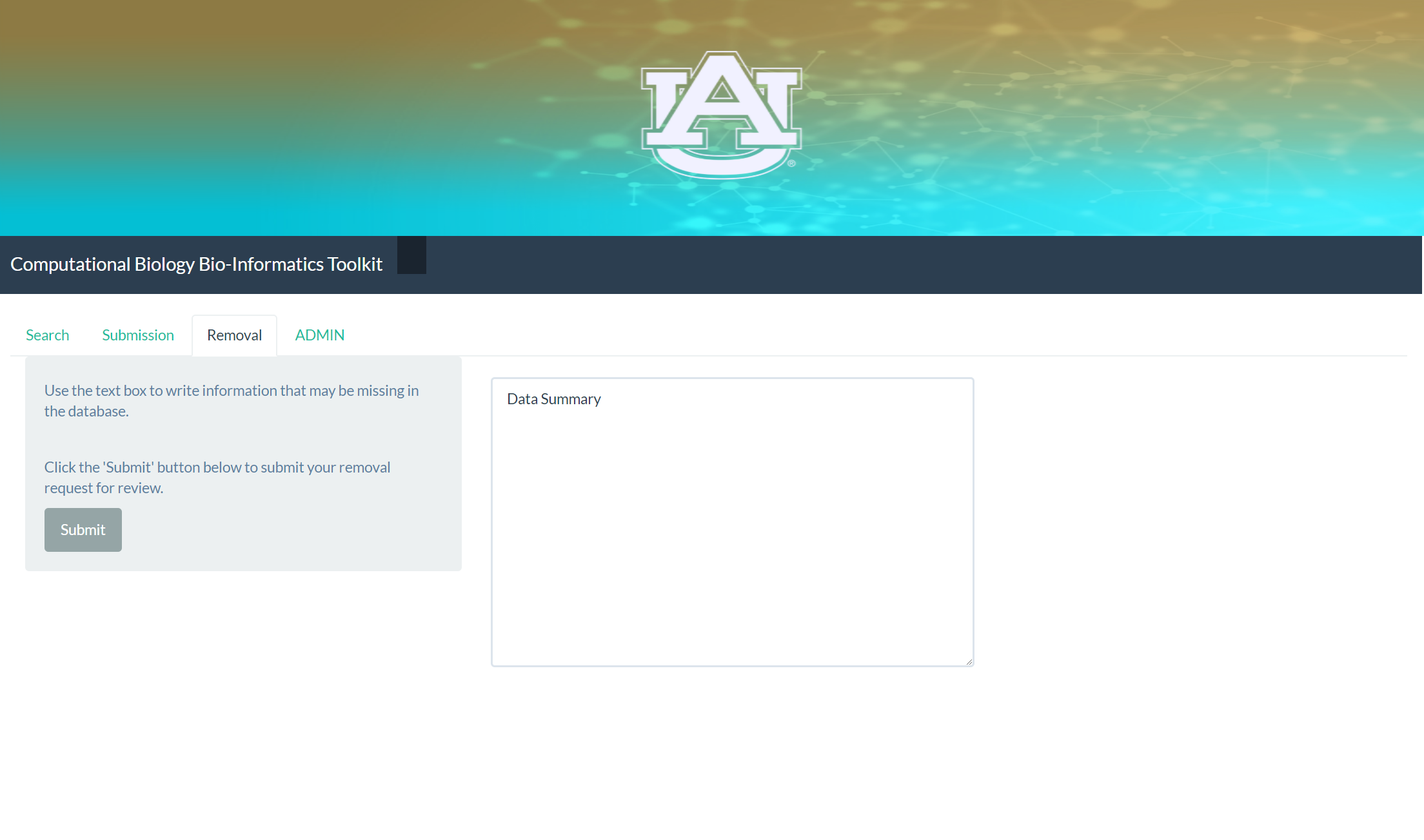
* HOME AND SEARCH PAGE:



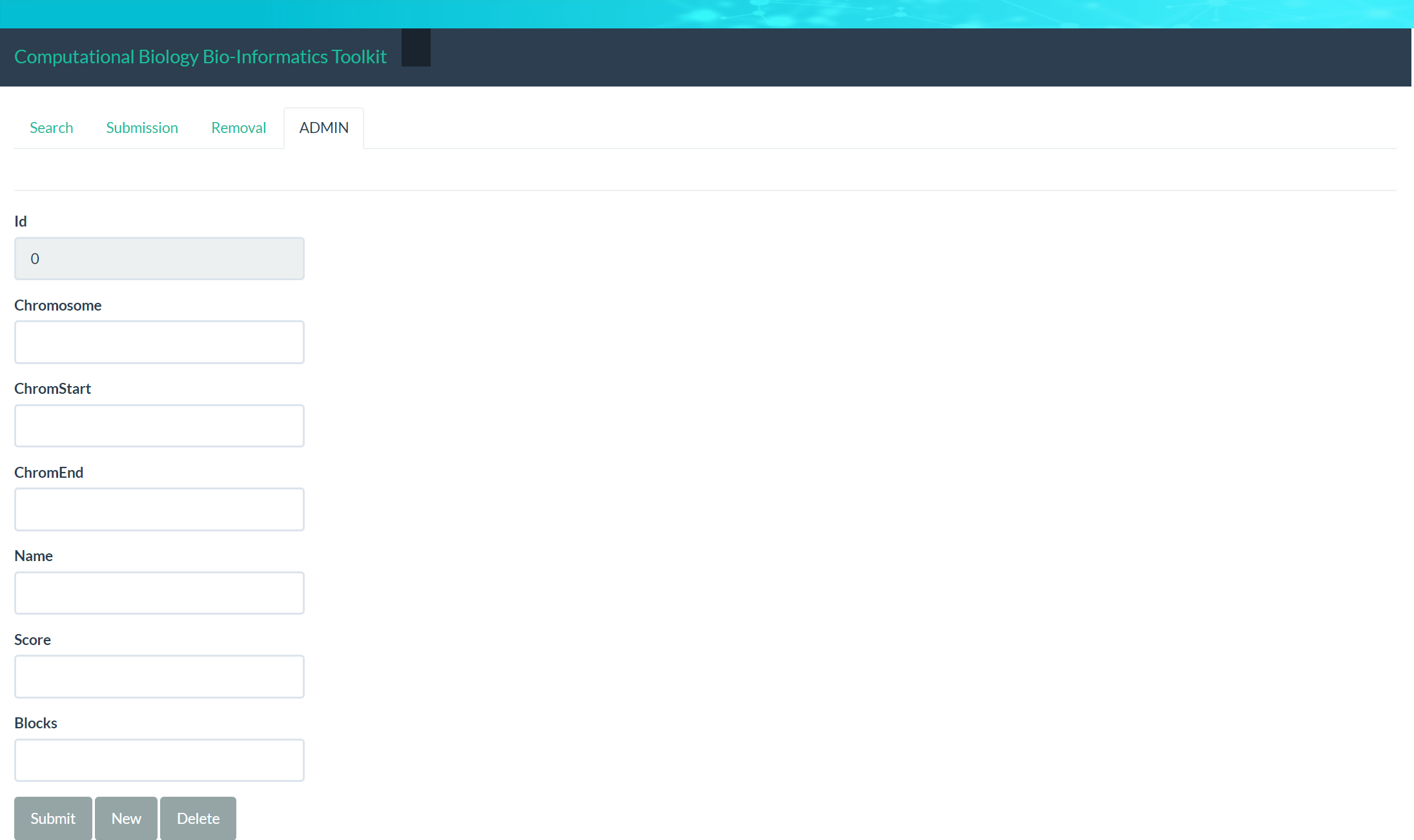
* USER DATABASE ENTRY REQUEST PAGE:



* USER DATABASE DELETION REQUEST PAGE:



* ADMINISTRATOR DATABASE PAGE:



*By: Ansleigh Yancey*

**5.2 User Requirements**

INPUTS

*Search*: The inputs are the drop down boxes for disease type and tissue and cell lines. These inputs determine the resulting genome signal information. Selecting the format type is an input for downloading results.

*Submission:* The inputs are text boxes for data submission and removal suggestions and file upload capabilities for data submission suggestions.

*Administrator*: The inputs are text boxes entries for the chromosome, start, end, name, score, and strand of the genome signal that the administrator has the authority to add or delete from the database.

* OUTPUTS

*Search*: the output is the list of genome signals that correlate with the disease type and cell line options chosen by the user.

*Submission:* There are no outputs for the delete and submission requests.

*Administrator:* For the administrator’s portal, the output for deletion and addition are direct changes to the database.

*By: Ansleigh Yancey*

6. MANAGEMENT PLAN

horizontal line

Tasks that are under development and being heavily prioritized are:

* Creating functions so that users are able to upload large datasets and see that the data is properly sorted/manipulated when processed within the web application
* Create algorithms that correctly and efficiently extract pieces of data from chromosomes and mutation data via indices so that they can be reproduced at will by a user within the web application when a certain category and mutation are both chosen
* Refine the User Interface further and link the front-end source code to back-end development efforts so that user testing can begin

**6.1 Task Assignments**

* Paul Ryu:
  + Finish coding and testing the uploading and organizing of data files when the user submits new information to the web application.
  + When the upload function is completed, move on to sorting chromosome data by their indices or positions.
* Benjamin Williams:
  + Complete the search functionality by merging relevant columns in the permanent dataset and the dynamic dataset. This will be done based on the positions of nucleotides on a chromosome within each dataset.
  + Display the merged data in a large table when certain traits are specified by the user.
* Sadaira Packer:
  + Make the dataset searchable by traits and mesh categories by linking the data from the trait dataset with the data from the cell line dataset.
* Ansleigh Yancey:
  + Make variable input options for trait drop down box correspond with the option selected in mesh traits drop down box
  + Make variable input options for mesh trait drop down box correspond with the option selected in traits drop down box
  + Work on finding the aspects of genome signals that correspond with the items selected by the user
* Mason Monday:
  + Finalize needed data points for our search functionality alongside our sponsor
  + Perform needed testing on current search algorithm so it can operate properly when our data finally gets added

**6.2 Development Schedule**

Key dates:

* April 01, 2019: Cycle 2 Report Due
  + *Plan*: Complete preparation for the presentation and continue refining the product so that a good demonstration can be shown to the class. Finish the ‘Upload dataset’ functionality and prioritize creating a proper sorting algorithm for mutation and chromosome data so that we can draw correlations between datasets as the client desires.
* April 05, 2019: Presentation Day, End of Cycle 2
  + *Plan*: Re-group after giving the presentation and getting feedback from Dr. Qin and the client.
* April 08, 2019: Cycle 3 Begins
  + *Plan*: Continue to refine any created algorithms for manipulating the datasets. If this key aspect is finished, we will then be able to do thorough testing with datasets that the client may have and move on to focus on the database and web server portion of the project.
* April 26, 2019: Presentation Day, End of Cycle 3 and the Project
  + *Plan*: If the project is completed and thoroughly tested for correctness and efficiency by this date, then refactor code for readability and prepare for final submission to the client and the database repositories (medical journals) that the client wishes to publish to. If the project is not completed by this date, write elegant documentation and create a report that includes every bit of helpful information so that either the client or groups in future semesters can utilize it to complete the project.
  + **IMPORTANT: CODE AND FEATURES WILL FREEZE AT THIS POINT IN ORDER TO ENSURE THAT WE HAVE ENOUGH TIME TO IDENTIFY AND FIX REMAINING BUGS DURING THE MOST RECENT DEVELOPMENT EFFORTS.**
* May 05, 2019: End of Auburn University’s Spring 2019 semester
  + *Plan*: Regardless of the outcome of the project’s completion, ensure that the client has everything that we have used and worked on the project with and thank him for allowing us to contribute to a very important field of computational biology and informatics.

*By: Paul Ryu*

7. TESTING PLAN

horizontal line

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Required Actions** | **Expected Results** | **Comments** | **✔/X** |
| Add Entry Request  (Text Box) | Navigate to the webpage. Choose the “send add request” option. Enter information on a genetic mutation or genome signal into the provided text box. The request to add the entry to the database is sent to the administrator. | Entry request sent to administrator |  |  |
| Add Entry Request  (Text File) | Navigate to the webpage.  Chooses the “send add request” option.  Upload a file with information on a genetic mutation or genome signal.  Request to add the entry to the database is sent to the administrator. | Entry request sent to administrator |  |  |
| Add Entry via Textbox (Admin) | Navigates to the admin webpage. Chooses the “add entry” option. Enter information on a genetic mutation or genome signal into the provided text box. | Entry added to database |  |  |
| Add Entry via File (Admin) | Navigate to the admin webpage.  Choose the “add entry” option.  Upload a file with the information on a genetic mutation or genome signal. | Entry added to database |  |  |
| Delete Entry Request | Navigate to the webpage.Find the entry that you would like to be removed. Choose the “send delete request” option. Enter reasoning for deletion (optional). The request to delete the existing entry is sent to the administrator. | Entry deleted from database |  |  |
| Delete Entry (Admin) | Navigate to the webpage.Find the entry that you would like to be removed. Choose the “delete entry” option. | Entry deleted from database |  |  |
| Edit Entry Request | Navigate to the webpage. Find the entry that you would like to edit. Choose the “send edit request” option. Enter the requested changes and reasoning for said changes in the provided text box. The request for the existing entry to be edited is sent to the administrator | Request sent to administrator |  |  |
| Edit Entry (Admin) | Navigate to the admin webpage.  Find the entry that they desire to edit.  Choose the “edit entry” option.  Alter the existing entry information using the text box. | Change is made to database entry |  |  |
| Search for Database Entry (cell line) | Navigate to the webpage. Complete the search entry based on predefined search options in the “cell line” drop down box. Choose the “search database” option. | The results matching the search criteria are displayed |  |  |
| Search for Database Entry (disease type) | Navigate to the webpage. Complete the search entry based on predefined search options in the “disease type” drop down box. Choose the “search database” option. | The results matching the search criteria are displayed |  |  |
| Search for Database Entry (cell line & disease type) | Navigate to the webpage. Complete the search entry based on predefined search options in the “cell line” and “disease type” drop down boxes. Choose the “search database” option. | The results matching the search criteria are displayed |  |  |
| Download Search Results | Navigate to the webpage.  Complete the search entry based on predefined search options in the “cell line” and/or “disease type” drop down boxes. Choose the “search database” option. Choose the download type (excel, text, word document). | A file is downloaded containing the results matching the search criteria |  |  |

*By: Sadaira Packer and Paul Ryu*

8. RISK MITIGATION

horizontal line

**Risk #1:** While writing our code for manipulating and combining the datasets, we may deviate from the final goal of the project if we do not follow the client’s needs with precision. Since many portions of the data are out-of-scope and cryptic to us as a group, this is a very possible risk since we may manipulate data in ways that are erroneous to the task at hand.

**Mitigating Risk #1:** We will closely communicate with the client throughout the next cycle so that we know that we dealt with each obstacle correctly and efficiently.

**Risk #2:** We have been at stasis because of a giant roadblock when trying to decrypt the data for the past three weeks and it is entirely possible that it will occur again as we delve deeper into the complexities of the assignment.

**Mitigating Risk #2:** Like how we solved this problem before, we must again be very aggressive when it comes to scheduling meetings with the client and work face-to-face so that we can clarify ambiguous information and identify exactly what our next objectives are.

**Risk #3:** Project may not be completed by the end of the semester.

**Mitigating Risk #3:** Back-end coding, the largest and most important portion of this project, must be accelerated greatly and the entire team needs to come together to collaborate on its algorithmic construction.

*By: Paul Ryu*

9. LESSONS LEARNED

horizontal line

**Ansleigh Yancey:** With cycle 2, I gained a better understanding of variable drop down box functionality in R and Shiny. I also gained a better understanding of how to manipulate the data based on options selected. I did more research into the aspects of the genome signals that are related to traits and mesh traits, and I brainstormed ways to use this information to populate search results in the next cycle.

**Mason Monday:** This cycle saw a great amount of my time parsing over data and testing for our search functionality. Ben and I spent a number of meetings with Dr. Chen discussing needs for searching, and we managed to finally narrow down which specific data points he wanted us to use. With this cleared-up information, we can finally break through our roadblock and continue with the project. This scenario shows how important pursuing information from our sponsor truly is, and what it takes to receive the knowledge we need from someone outside of our field of study.

**Benjamin Williams:** During the Cycle 2 phase, I primarily developed my skills in R Shiny regarding data organization and manipulation across multiple datasets. I worked on both the front-end and back-end, and gained a lot of useful database experience in the process.

**Paul Ryu:** During Cycle 2, we were able to extract much more information from the client so I learned many things about the seemingly hieroglyphic data that we were initially given. The final objective is clearer to me and development has been easier as a result this time. However, it is still a very big problem that we are only four weeks out from the end of Senior Design so I would say that the biggest lesson I have learned overall is that I need to be much more aggressive in meeting with clients in the future because it can result in a lot of wasted time and effort if not.

**Sadaira Packer:** Looking at other cell line databases wasn’t very helpful in helping us combine the datasets. There being multiple cell lines for some of the dataset entries will complicate the search algorithm. If there was only a single cell line per entry we could display results by searching the cell line column for matching variables. But because some variables in the cell line column have multiple cell lines, this won’t display all the proper results.