

TABLE OF CONTENTS

1. System Metaphor ………………………………………………………………………………………………… 2
2. Design Overview ………………………..………………………………………………………………………… 3
   1. Background Information
   2. System Evolution Description
   3. Constraints
3. System Architecture.……………………………………………………..……………………………………… 4
   1. Hardware Architecture
   2. Software Architecture
4. Data Design …………………………..……………………………………………………………………………… 5
   1. Database Management System Files
   2. Non-Database Management System Files
5. Software Detailed Design ……………………………………………………………………………………… 6
   1. User Stories
6. User Interface ………………………………………………………………………………..…………………..… 14
   1. Inputs
   2. Outputs

1 SYSTEM METAPHOR

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, 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.

2 DESIGN OVERVIEW

* 1. Background Information
* Our client, Dr. Li Chen, is facing an issue where there is a lack of methods and research for organizing diseases, tissues, and cell types for the human body. We have been given two datasets. One dataset, an online database hosted by the National Center for Biotechnology Information, contains information about genetic mutations for diseases. The other dataset was given to us in the format of a ‘.bed’ filetype and contains human genome signal data from the tissues and cells. Our goal is to link these two datasets together so that we can see which genome signals occur more in specific genetic mutations.
  1. System Evolution Description
* Currently, our team is working toward creating a user interface that satisfies the request of our client. We are learning the capabilities of R in regards to databases and learning the functionality of Shiny in regards to web application.
  1. Constraints
* The members of our team, prior to working on this project, have not had a lot of practice with databases. The major constraint of this project is our overall lack of experience, but we are willing and excited to learn and have already made great strides.

1. SYSTEM ARCHITECTURE

3.1 Hardware Architecture

* Our team is using the Windows Operating System with the following processors:
  + Paul Ryu - Intel i9-9900k, 5.0GHz
  1. Software Architecture
* We are using R in the RStudio environment.
* Application development will be done using Shiny, R’s web application framework.

1. DATA DESIGN

4.1 Database Management System Files

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | chrom | chromStart | chromEnd | name | score | blocks |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |

* 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
* Documentation on the table entries can be found at https://genome.ucsc.edu/FAQ/FAQformat.html#format1

4.2 Non-Database Management System Files

* 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

1. SOFTWARE DETAILED DESIGN

3.1 User Stories

**User Story:** Add Entry Request (Text Box)

**Brief 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.

**User Story:** Add Entry Request (Text File)

**Brief 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.

**User Story:** Add Entry Via Text Box (Administrator)

**Brief 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:**

1. The Entry is added to the database

**User Story:** Add Entry Via File (Administrator)

**Brief 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:**

1. The Entry is added to the database.

**User Story:** Delete Entry Request

**Brief 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.

**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.

**User Story:** Delete Entry (Administrator)

**Brief 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.

**Post-Conditions:**

1a. The Entry is deleted from the database:

* The administrator removed the desired entry from the database.

**User Story:** Edit Entry Request

**Brief 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.

**User Story:** Edit Entry (Administrator)

**Brief 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 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.

**User Story:** Search for Database Entry (cell line)

**Brief 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.

**User Story:** Search for Database Entry (disease type)

**Brief 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 for the disease type chosen by the actor do not yield a result.
* There is no existing entry in the database.

**User Story:** Search for Database Entry (cell line & disease type)

**Brief 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.

**User Story:** Download Search Results

**Brief 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).

**Post-Conditions:**

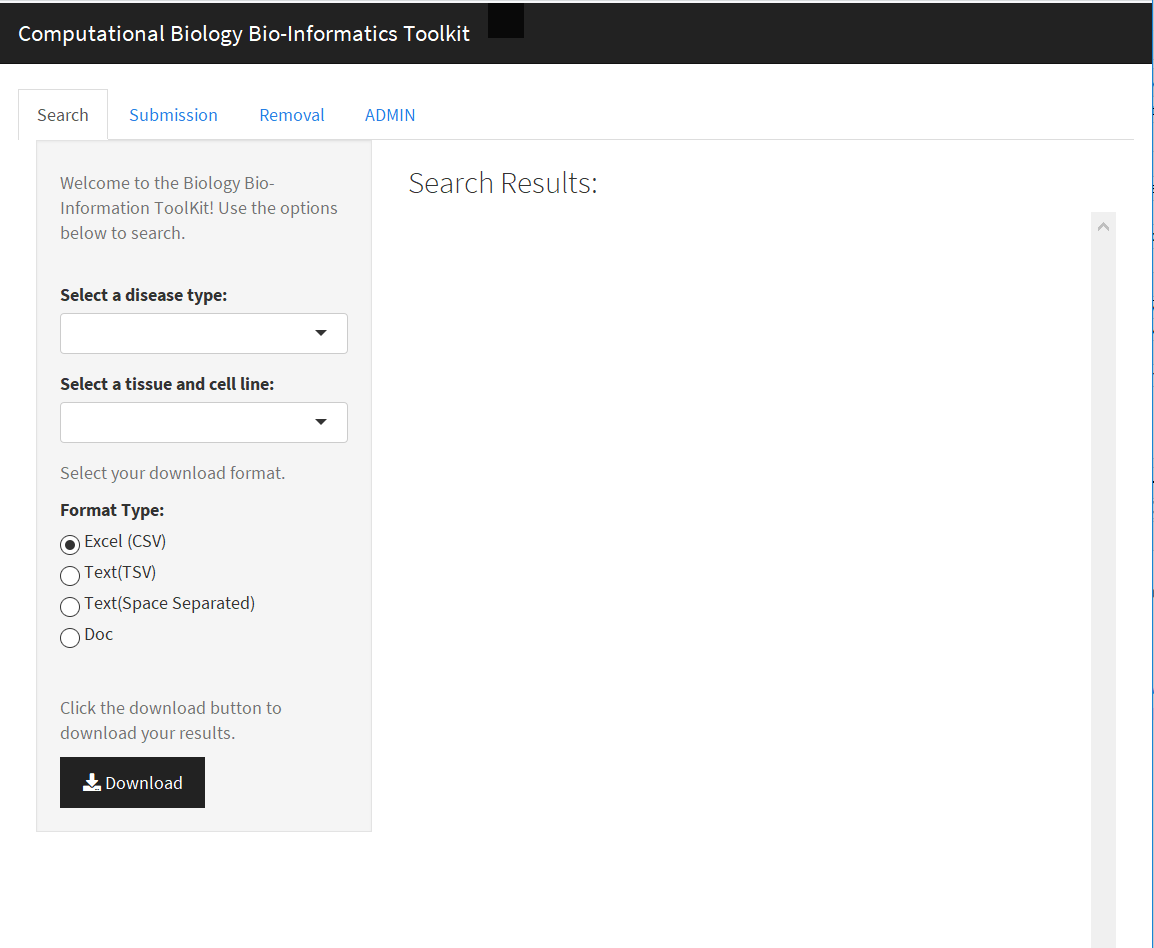
1. The Existing Entry is Found:

* The actor saves the file to his/her personal device.

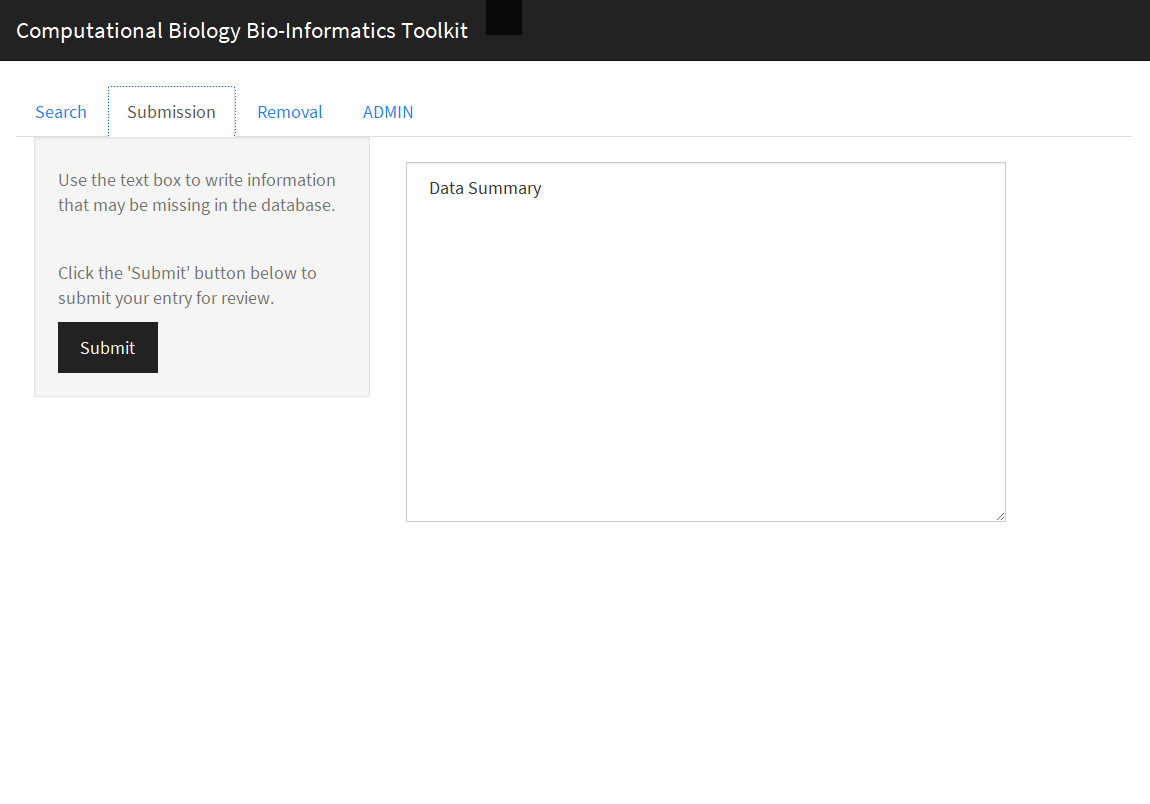
1. USER INTERFACE

6.1 Inputs

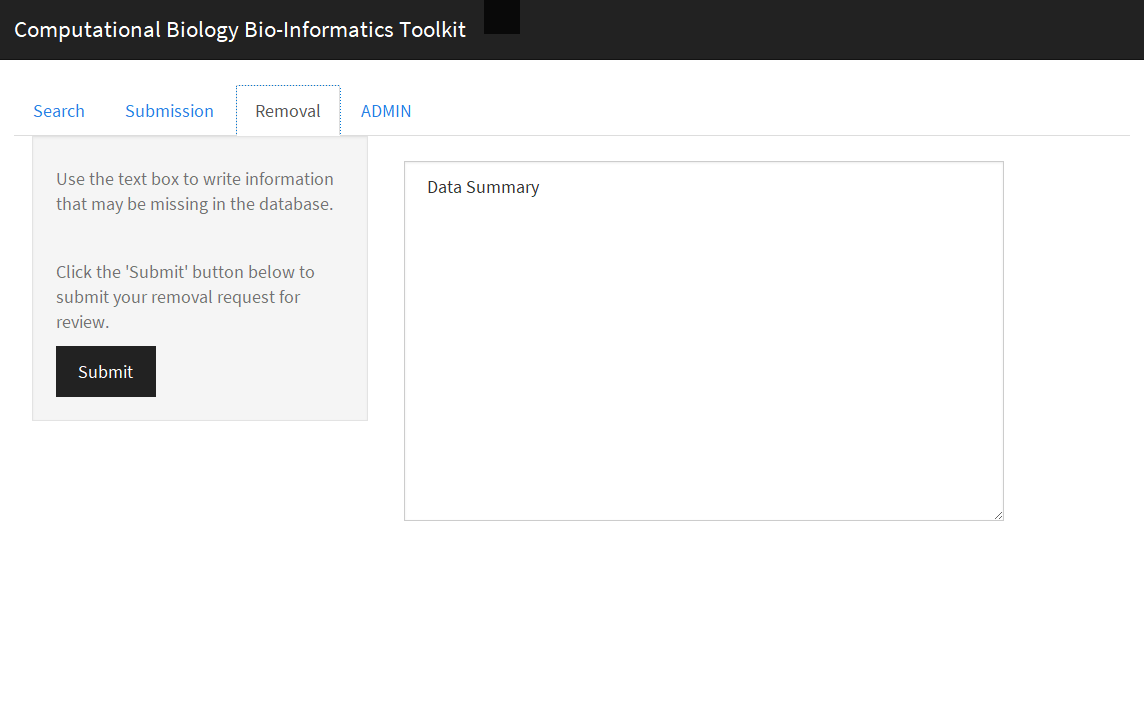
Search Inputs:



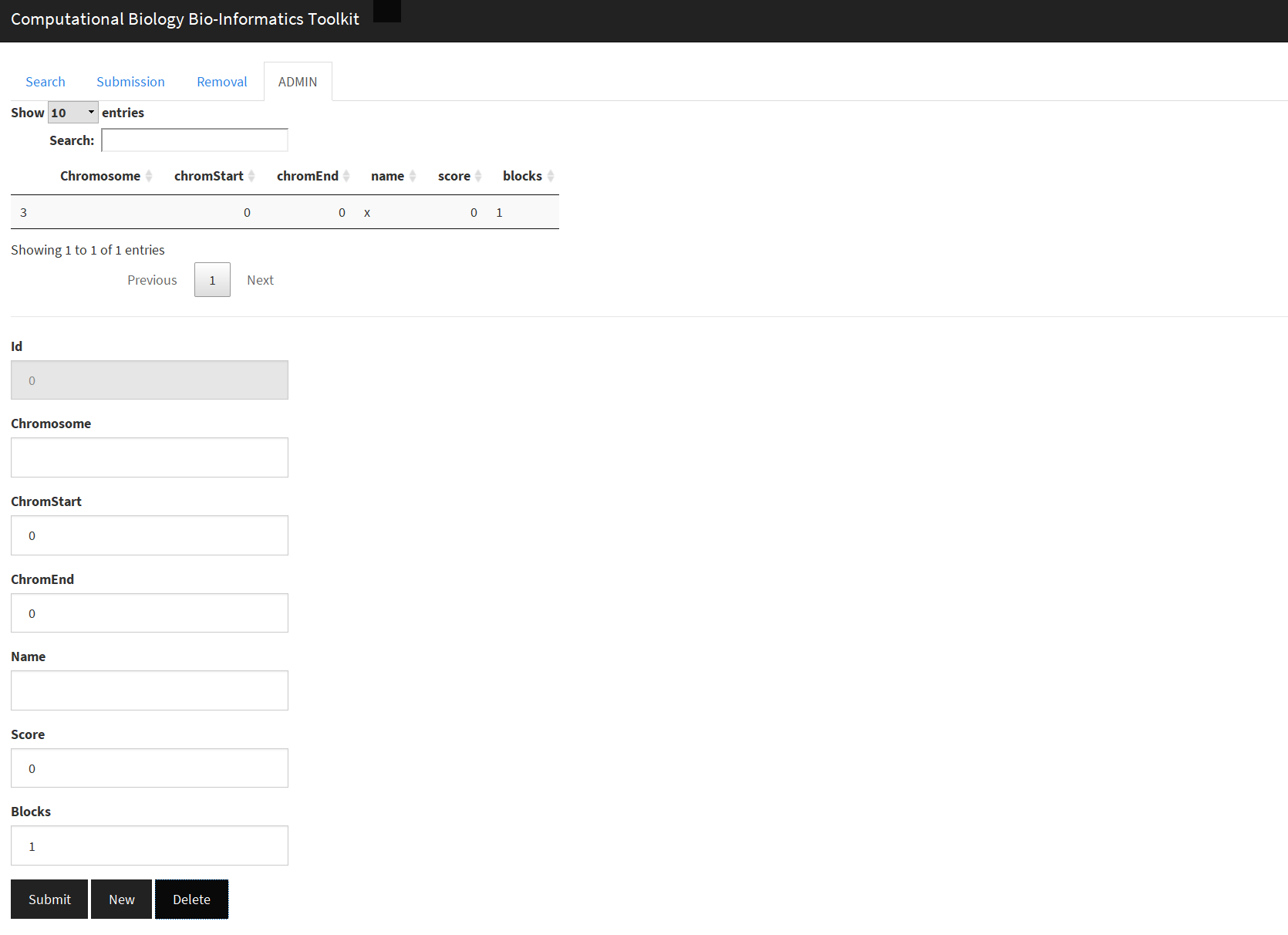
The Image above is a screenshot of the User Interface for the Search portion of our web application. The inputs are the drop down boxes for disease type and tissue and cell lines. These inputs determine the resulting genome signal information. Selecting the format type is an input for downloading results.



The Image above is a screenshot of the User Interface for the Submission portion of our web application. The input is a text box for data submission suggestions.



The Image above is a screenshot of the User Interface for the Removal Request portion of our web application. The input is a text box for data removal suggestions.



The Image above is a screenshot of the User Interface for the administrator’s portal portion of our web application. 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.

6.2 outputs

* For the search tab of our application, the output is the list of genome signals that correlate with the disease type and cell line options chosen by the user.
* There are no outputs for the delete and submission requests.
* For the administrator’s portal, the output for deletion and addition are direct changes to the database.