CSC 366
Winter 2012
Course Project

Regulatory Sequences Database

Project Documentation
Use Cases

Version 2.0

Dan Peterson
Alyssa Conly
Laura Strand

Alex Dekhtyar
1. General Information

This document outlines the use cases for the database of regulatory sequences found in the promoter regions of up- or down-regulated genes discovered when comparing two states of a cell to each other.

In what follows, the term *database* is used to refer to *both* the collection of data described in the companion document, *and* to the software system you are being asked to design and develop in this course. Where necessary, we disambiguate by referring to the latter as the *database software* or the *database application*.

There are two ways to implement the database software: as a stand-alone application that connects to a local database (or a database on a local network) for data, and as a web-based application that runs inside the browser. Except for some user management issues, discussed at the end of this document, all other use cases do not differentiate between the two approaches. The database software is to be designed as essentially a single-user system. The necessary on-line precautions are discussed at the end of the document.

The database functionality is separated into three broad categories:

- Data insertion/manipulation functionality;
- Data browsing/filtering functionality;
- Analytical functionality.

**Data insertion** functionality consists of use cases for inserting the data into the database, as well as the functionality for modifying the data that is already in the database.

**Data browsing and filtering** functionality consists of use cases for showing the users the contents of the database in a convenient form.

**Analytical functionality** consists of use cases the users of the database plan to use in their research pursuits.
2. Data Insertion and Manipulation Functionality

[UC-1.] **Bulk Insertion/Insertion of data for a single experiment.** Most of the time, when the database is updated in a significant way, the events immediately preceding the update involve a researcher/database user performing an in-vitro experiment comparing two cell states, determining the proteins that were different in abundance, running TESS queries for all discovered proteins/genes and collecting TESS answers in a group of spreadsheets residing in a single directory.

Because of this, the core database update/data insertion functionality is the **bulk insertion** of the data for one experiment or more experiments (if data for more than one experiment is available at the same time).

The data to the database will be provided in a collection of CSV files representing individual worksheets of the Excel spreadsheets in which TESS returns the data.

**Information provided.** The user selects the bulk update functionality from the main menu/main screen of the database software. The software provides the **bulk insertion** dialog. The user selects a local directory that contains the CSV files for the experiment. This action is subject to the following restrictions and specifications:

- The directory will contain four CSV files for each discovered in the experiment up- or down-regulated gene. The four files storing information about the same gene will have similar filenames, different only by the designation of the worksheet from the Excel file, which is used. The four types of CSV files are:
  1. *Experiment description (Job parameters).* This CSV file will contain <Name, Value> pairs describing the experiment and the gene.
  2. *Sequence.* This CSV file will contain the 2000 character promoter region sequence for the gene.
  3. *Regulatory sequence discoveries.* This CSV file will contain information about discoveries of individual regulatory sequences in the promoter region of the gene.

- The filename conventions are specified in the **Lab 5** assignment.
- In the past CSC 366 course we discovered that some software development frameworks used by students in the course lacked the predefined facilities for selection of a single directory (rather than selection of a file, or multiple files). Please know that it is highly desirable for the software for the users to be able to select directories, not files.

**Software actions.** Upon selection of the directory, the software shall scan the CSV files in the directory. The software shall create a list of gene names, the information about which is contained in the CSV files. For a gene name to be included on the list, all four CSV files for this gene must be present in the directory. The software then, displays the list of genes to the user. The user inspects the list of genes, and is provided with the opportunity to exclude individual genes from the upload. Upon completion of the gene selection, the user presses the **Upload** button. The software reads, one by one, the files for each gene, and updates the database with the information on the experiment, the gene, its promoter region, the regulatory sequences found in the promoter region and the
transcription factor summaries. The software may provide the user with some feedback concerning the ongoing insertion activities, as the bulk upload proceeds.

**Note.** This use case applies to two situations: a *bulk upload proper* used to populate an empty database, and an *addition of a new experiment* to the database. In the former case, the database is empty, and the data to be inserted contains, potentially, multiple experiments. In the latter case, the data for a single experiment becomes available and is uploaded in bulk. The system behavior should be the same in both cases, except for «noticing» how many different experiments are being uploaded.

[UC-2.] **Loading information about a single gene.** To make it easy for the users to compensate for errors of omission (should any Excel spreadsheets from which the CSV files are constructed become corrupt), this use case offers the users of the system the opportunity to upload information about a single gene found up- or down-regulated in a *specific experiment*. The uploaded data is available in four CSV files.

**Information provided.** The user selects «Upload Single Gene Data» functionality from the main menu/screen of the software. The system brings up the «Upload Single Gene Data» dialog screen. On the screen, the user navigates to a directory that contains the files for upload. When the directory is selected, the software reads the list of CSV files in the directory and determines the names/labels of all genes, whose complete descriptions are found. A description is complete if all four CSV files describing the data to be stored in the database are found. The software outputs the list of available experiments and gene data to the user. The user selects the gene name the data for which (s)he wants to upload and presses the «Upload» button (or initiates the upload using some other interaction with the dialog screen).

**Software Actions.** The software reads the contents of the four CSV files associated with the gene selected by the user, verifies the information about the cell comparison and the experiment (and if they are not represented in the database yet, inserts this information) and uploads the information about the gene, its promoter region, discovered regulatory sequences and transcription factor summaries to the database.

[UC-3.] **Editing the contents of the database.** The software shall provide the facilities for updating individual records stored in the database. In Section 3 of this document, the browsing functionality is detailed. As part of the browsing functionality the user shall be given the ability to view individual records. It is convenient to incorporate the *update record* functionality and *delete record* functionality together with the functionality for viewing records.
2. Browsing and Filtering Functionality

[UC-4.] Hierarchical browsing of the database. This functionality can be implemented on either the front page of the database application, or on a page that is easy to access from the front page.

The key hierarchical organization of the data in the database (as described in the prior documentation) is as follows:

- The researchers are interested in comparing two different states of the same cell from a specific specie.
- For each such comparison, researchers can conduct multiple experiments.
- With each experiment, a collection of genes found to be different in abundance in the two cell states is associated.
- For each gene identified this way, a list of regulatory sequences and associated factors found in its promoter region is stored.

One of the main mechanisms for study of the data by the users revolves around browsing the data in this hierarchical manner.

User-software interaction.

- The user selects to pursue the hierarchical browsing functionality of the software (if the hierarchical browsing is enabled on the front page of the software, then what follows occurs without the user making an explicit selection of activity).
- The software displays a list (or a table) of species, currently available in the database.
- The user selects the specie of interest.
- The software, while keeping the list of species on-screen, produces a list of comparisons, currently available in the database. For each comparison, include the name of the specie for which it is made.
- The user selects the comparison of interest.
- The software, while keeping the list of the comparisons on-screen, produces a list of available experiments for the given comparison. The information displayed within “Experiment” includes date of the experiment and the name of the person who staged it, the number of genes identified, the number of genes showing “Regulation” up or down.
- The user selects the experiment of interest.
- The software, while keeping the list of species, comparisons and the list of experiments on-screen, produces the list of all genes found different in abundance for the experiment. For each gene, its name, location (chromosome, begin site, end site), whether it is up- or down-regulated, and the total number of discovered factors.
- The user selects a gene of interest.
- The software, while keeping all previous lists on-screen produces the list of all factors found in the promoter region of the gene. For each factor, its origin (model), the name of the factor, the number of times it is found in the promoter region.
- The user selects a factor of interest.
- The software, while keeping all previous lists on-screen produces the list of all regulatory sequences found for that factor in the promoter region of the gene. For each regulatory sequence, its position (beginning, length, sense), each “quality” measure (all the “l-values”) and the actual string shall be displayed.
• The user selects a regulatory sequence
• The software shows the user a dialog screen for viewing (as well as editing) information about a single regulatory sequence (see use case [UC-8]).

[UC-4.1] User filtering of the displayed information. While the user is perusing hierarchical browsing, the following filtering options shall be made available.

• For the list of genes for a given experiment, the software shall provide the user with the ability to select all genes, up-regulated genes only or down-regulated genes only.
• For the list of regulatory elements for a given gene (in a given experiment), the software shall provide the ability to
  o Sort the list by any displayed column
  o Filter the list by any of the l-values stored in the database. The user provides a range of values, or a lower bound on the specific l-value parameter (any single l-value parameter). The software keeps information about regulatory elements whose l-value is in the appropriate range.
  o Location on the promoter sequence. The user provides information about the range (“from” and “to” positions) for the beginning of the regulatory element and, optionally, select the sense of the regulatory element. The software keeps in the list only the regulatory elements with starting points in the specified range (and with the specified sense, if one is provided).

[UC-5.] Summary list. [DEPRECATD]

[UC-6.] Database Summary: Genes.
• The user selects “Database Summary: Gene list” from the main menu/main screen of the system.
• The software brings up a dialog screen and populates it with the list of genes found in the database. For each gene, the following information should be provided:
  o Species (i.e., the same gene found in different species should show up multiple times in the table)
  o Gene name
  o Gene abbreviation
  o Gene location (chromosome, begin site, end site)
  o Number of comparisons in which the gene was found different in abundance (all occurrences, down regulated occurrences, up regulated occurrences)
  o Number of experiments in which the gene was found different in abundance (all occurrences, down regulated occurrences, up regulated occurrences)

• The user can sort the genes by any column in the output.
• The user selects a gene of interest.
• The software opens a dialog screen containing the information about an individual gene. The new screen contains all information about the gene available from the database (name, location, etc), as well as the list of experiments in which the gene was found to be different in abundance as well as the regulation (up or down) for each experiment.
[UC-7.] **Database Summary. Regulatory elements/Transcription factors.**

- The user selects “Database Summary: Transcription factors” functionality.
- The software brings up a dialog screen and populates it with the list of all transcription factor names found in the database. For each transcription factor, the list contains the following information:
  - Transcription factor name
  - “FAC” (if there is one present)
  - Number of models in which this transcription factor was found anywhere in the genes of interest (i.e., those that were different in abundance in any of the stored experiments)
  - Number of genes in which this transcription factor was found
  - Total number of occurrences of the transcription factor in the database
- The user can perform the following manipulations with this list:
  - Select a transcription factor of interest. In response, the software brings up a dialog screen populated with the list of all occurrences of the given transcription factor in the regulatory elements in the database. The list can be sorted by each column it contains (comparison name, experiment information, gene name, regulatory element location, model number).
  - Sort the list of transcription factors by any column in the list, in particular by the number of models, the number of genes and the number of occurrences.

[UC-8.] **Viewing information about an individual regulatory element for a gene.**

At the bottom of many navigational hierarchies in the system lies the dialog screen that shows the information on a single regulatory element for a given gene from a given experiment. The dialog screen showing this information shall contain the following:

- The information about the comparison and the name of the experiment.
- The information about the gene for which the regulatory sequence is found.
- The information about the regulatory element itself:
  - Its location in the promoter sequence (beginning, sense and length);
  - The nucleotide sequence;
  - The origin of the regulatory element discovery (model);
  - The quality of the match (all the l-values).
- The name of the transcription factor(s) associated with the sequence

Additionally, the dialog screen shall provide for the following types of interaction with the user:

- Editing the information about the regulatory element.
- Deleting the information about the regulatory element.
- Searching for similarly described/duplicate regulatory elements in the given promoter sequence. Searching for this information shall result in a list of similar regulatory elements displayed on the dialog screen.
  
  A regulatory sequence is **similar** to another regulatory sequence in the same promoter region if both sequences start at the same position, go in the same direction and their lengths are within 1 nucleotide of each other.
3. Analytical Functionality

Note: UC-4.1, User filtering function would be very helpful to apply to all the UCs within the analytical functionality. I.e. allow the user to filter all output according to the specifications in UC-4.1. Part of this has been added as specific components of UC-9 and UC-10.

[UC-9.] Finding all genes that have specific transcription factor or factors present.

One of the key purposes of this database is to allow the researchers to study similarities between different genes found to be important for a comparison of two states of a cell. One of the key analytical tasks is to find and report a list of genes that contain a specified single transcription factor or all specified transcription factors from a given list.

**Inputs provided by the user.** The user supplies the following information to the software:
- The transcription factor or factors of interest.
- Constraints for quality parameters for factors (L-values) that, if provided by the user, limits the analysis to factors that conform to the constraints.
  - Minimum L a
  - Minimum L a/
  - Minimum L q
  - Maximum L d
- The scope of the search. The scope of the search can be an entire database, a single specie, a single comparison, or a single experiment.

The software shall provide convenient, user-friendly UI for supplying the input information.

**Information provided by the software.** Upon accepting the inputs provided by the user the software searches the specified part of the database for genes, whose lists of discovered regulatory elements contain elements to which all transcription factors from the input list bind. For each gene, the software shall return the following information:
- Gene name
- Comparisons in which the gene was identified and Regulation (UP or DOWN) for each comparison
- Experiment for each comparison

The software shall provide the opportunity to reorder the output by the contents of any column.

[UC-10.] Listing transcription factors by popularity in the database.

Another key purpose of this database is to allow the researchers to identify factors that are centrally involved in the comparisons being studied. This can be accomplished by allowing the researchers to select multiple comparisons (or all comparisons) and listing all of the factors that appear in any of the genes identified from those comparisons, sorted by number of appearances.

**Inputs provided by the user.** The user supplies the following information to the software:
- Species (can be one or multiple)
For each species selected, a list of all available comparisons appears, and user selects one or more (up to all) comparisons.

For each comparison, if there is more than one experiment, the user can select one or more experiment.

Constraints for quality parameters for factors (L-values) that, if provided by the user, limits the analysis to factors that conform to the constraints.
  - Minimum L_a
  - Minimum L_a/
  - Minimum L_q
  - Maximum L_d

The software shall provide convenient, user-friendly UI for supplying the input information.

Information provided by the software. Upon accepting the inputs provided by the user the software searches the specified part of the database for factors that appeared in any gene included in the user inputs, and provides a list of all such factors, ordered by number of times the factor appeared in the portion of the database included in the search. The resulting list includes:
  - Factor (starting with most “popular” or highest number of appearances)
  - Number of times the factor appeared
  - Number of genes in which that factor appeared
  - When the factor is chosen, list of genes in which the factor was identified
  - For each gene, comparison(s) and experiment(s) in which the factor was identified

[UC-11.] Listing factors found in specific regions of the promoter. [REMOVED]

[UC-12.] Listing genes, which have transcription factors in specific regions. [REMOVED]

[UC-13.] Finding genes that share multiple transcription factors [EXTRA CREDIT].

Explanation. This is an EXTRA CREDIT use case. It involves implementation of the Apriori algorithm (or any other algorithm for computing the list of frequent itemsets).

The nature of the task is as follows. With each gene stored in the database, a list of transcription factors found to bind to its promoter region is associated. It is of interest to the users of the system to be able to find the lists of transcription factors that are commonly shared, and be able to view the list of genes that share these transcription factors.

An itemset as applied to this task, is a collection of transcription factor names (associated with a single gene). Given a list of genes, the support of an itemset of transcription factors is the percentage of the genes, which contain the given itemset of factors.
An itemset is *frequent* if its support is above some value \(0 \leq a \leq 1\), specified by the user. The goal of this use case is for the software to determine the list of frequent itemsets in your database, and to provide the list of applicable genes for each frequent itemset.

An itemset is called a *skyline itemset* if it is frequent by has no frequent supersets. The software should report only *skyline itemsets* (lest, there may be way too many itemsets to report).

There is a number of algorithms for discovery of frequent itemsets, most of the relying on the *Apriori principle*:

\[\text{Any subset of a frequent itemset must be frequent.}\]

**Inputs provided by the user.** The user will provide the software with the following information:

- The filter on the list of genes to be searched (all genes, genes from a specific experiment or a collection of experiments)
- The *frequency threshold* \(a\): a number between 0 and 1, representing the percentage of all genes in which an itemset must appear to be reported, or a *minimum number of occurrences threshold*: a number of genes in which the itemset is to appear to be reported.
- Itemset size threshold: the smallest possible size of a reported frequent itemset.

**Information provided by the software.** Given the parameters above, the software shall implement the Apriori algorithm (see below), or any modification of the Apriori algorithm deemed sensible by the development team to find the list of all *skyline frequent itemsets* that match the frequency threshold and itemset size threshold.

The frequent itemsets shall be reported in a convenient succinct manner. Selection of a frequent itemset shall lead to a dialog window which shows the list of all genes that have all the transcription factors from the selected itemset associated with them.

**[UC-14.]** Sort lists of transcription factors by "quality". [REMOVED]

**[UC-15.]** Promoter Region Map [EXTRA CREDIT].

This is an *EXTRA CREDIT* use case. Bioinformatics databases that store genomics data often display information about various places of interest (genes, regulatory sequences, etc...) using a *genomic map*. A genomic map in this context, is a representation of a linear DNA sequence using UI or graphical elements, that shows, via color-coding, and/or other means, where on the observable part of the sequence various "interesting" places lie. The map is usually responsive to user interaction: e.g., a user can click a mouse button on a highlighted location to receive more information about it.

For the software system you are building, a map can be created for the promoter region sequences. A promoter region is associated with a single gene of a specific species, so, a promoter region map can be displayed and used for interactive purposes whenever information about regulatory sequences
of a single gene (from a specific experiment) is shown. Additionally, there may be value in showing multiple promoter region maps together, at it might provide some visual cues for the software users about the similarities between regulatory element occurrences in different promoter regions.

A promoter region map is essentially a color-coded, and/or otherwise marked up line (represented as a graphical element and/or in some other way), in which an internal mapping between each physical (pixel) position on the line and a location in the promoter region is established. (You can also represent the map as multiple lines: e.g., one for each DNA strand). The map shall, at least, contain references (color-coded regions, tick marks, or some other ways) to the location of different regulatory elements in the promoter region. The references shall be “clickable” in a sense that there should be a way to interact with the promoter region map and “select” a specific mark/region. Selection of such a region shall result in additional information (essentially a record, or records about the regulatory sequences associated with the selected location) becoming visible/available/highlighted/selected in the current dialog window.

Even a simple promoter region map may be of great use to the users of the system.