Anindita Varshneya

BIOL 598-04: Advanced Systems Biology Research

**Test-driven development and new features improve GRNsight: a web application and service for visualizing small- to medium-scale gene regulatory networks**

Outline

1. Introduction
   1. Gene regulatory network definition and use cases
   2. Typical presentation of network data is adjacency matrix
   3. GRNmap, GRN modeling and parameter estimation
      1. Produces weighted and unweighted adjacency matrices
   4. Visualization of GRN data allows for easier interpretation
   5. Importance of visualization systems
   6. Competitors such as Cytoscape, BioTapestry, etc.
   7. Niche that GRNsight addresses
   8. Basic features of GRNsight
      1. Uploads with Excel, GraphML, SIF
      2. Edge coloring and weights
      3. Draggable nodes
      4. Force Graph Parameters
      5. Print
   9. Sophisticated architecture
      1. Service-oriented architecture
      2. Server to web client relationship
   10. Open Source Development practices
   11. Compatibility
   12. Improvements required to develop GRNsight into a well-rounded program
       1. Testing to ensure accuracy of data
       2. Good responses to invalid adjacency matricies
       3. Additional front-end features to improve analysis of networks
2. Materials and Methods/ Results
   1. Testing Framework
      1. Created about 500 sample adjacency matrices to test for various different possible edge cases
         1. Included tests for gene names, adjacency matrix, data, formatting, and special characters
         2. All unit tests are for server side functions to ensure that output data is accurate and a good representation of adjacency matrix
         3. Initially, ran manual tests using these spreadsheets by uploading each sheet onto GRNsight and testing the functionality of the website
      2. Implemented unit testing framework using Mocha and Chai assertion library
         1. Created test functions according to different errors and warnings that GRNsight tests
         2. Test functions corresponded to particular errors and warnings that were checked for within the spreadsheet controller
         3. Test functions check for the frequency of certain error/warning codes within the error/warning array of the network object
         4. Each sheet produced in the previous section is sent through the
   2. Updated Warnings and Errors System
      * 1. Created distinction between fatal and non-fatal mistakes in adjacency matrix
        2. Created warnings functions and implemented checks throughout spreadsheet controller to ensure that all code is checked for non-fatal errors (i.e. warnings)
        3. Wrote descriptive warnings messages
           1. What the problem is
           2. Where the user can locate the issue in the Excel sheet
           3. How the user can attempt fixing the problem
           4. Consolidates warnings of similar types to prevent repetitive data/ information to the user
        4. Created an upper limit regarding how many warnings a sheet can produce before the sheet is “rejected” and returns an error
        5. Updated error messages to follow similar framework as warnings messages mentioned above
   3. Front-End Features
      1. Centered all arrowheads with the edges
      2. Allow weights to always show, never show, or show on mouse over
         1. Presents relationship between genes at the center of corresponding edge
   4. Other
      1. Created strictness parameter for how extraneous data is handled
         1. If extraneous data found within the strictness parameter, a warning is sent to the user
         2. If extraneous data is found outside of the strictness parameter, an error is sent to the user
      2. Began process of implementing graph statistics
         1. Explored Cytoscape.js library for graph statistics algorithms
         2. Began exploring the implementation of betweeness centrality and shortest path within GRNsight networks
         3. Ran tests and manually calculated the statistics in order to ensure accuracy
         4. Some questions raised regarding presentation of data and accuracy of shortest path considering directed graphs and self-regulating genes
3. Discussion
   1. Testing framework allows us to ensure our network visualizations are accurate
   2. Warnings and error systems ensure
      1. Users only upload documents that won’t crash the server
      2. Users get enough information about the problems with the network so they can fix it
   3. Front end features allow for users to easily use software
      1. Ensures the visualization is aesthetically pleasing
      2. Hiding/showing weights customizes visualization to accommodate different uses
   4. Handling extraneous data provides some flexibility, but ensures user isn’t uploading incorrect adjacency matrix
   5. Graph statistics provide more useful information about networks
      1. Allows for easier and quicker in-depth analysis and interpretation
   6. *Additional features incorporated by other researchers in the project?*
   7. Future Direction
      1. Currently working on customizing normalization factor
         1. Edge weights have thicknesses based on normalized data
         2. If data is normalized by different factors, their edge weights will not be the same
      2. Currently network graphs normalized according to minimum and maximum value in the graph itself
      3. Allows user to easily and directly compare two network graphs by inputting the maximum value of both graphs so both graphs are normalized by the same factor
4. Figures and Tables
   1. Weighted and unweighted adjacency matrix
   2. Server and web client relationship
   3. Overview of main components on GRNsight
      1. Perhaps use an edited version of annotated GRNsight image from poster
   4. Diagrams of possible edge types and their meanings
      1. Magenta, cyan, self regulation, arrow types, magnitudes
   5. Test coverage statistics
   6. TDD process graphic
   7. Network with all weights showing vs. network with weights showing on mouse over
   8. Warning/error modal with descriptive message
      1. Warnings modal drop down menu
   9. Addendum
      1. Current state diagram?