Jose Carrillo

Professor Ash Stuart

Advanced Bioinformatics

10 May 2019

Leader Genes Approach: An *ab initio* method for predicting functional genes of complex diseases

**Abstract:**

Genomic analyses are ineffective in accurately identifying genes involved in complex, multifactorial diseases due to the overwhelming amount of data they produce. Experimental lab analysis of individual or small groups of genes becomes increasingly costly and time-consuming. A bioinformatics *ab initio* method, the leader genes approach, was undertaken to lessen this data load while maintaining a high standard of accuracy; an exhaustive search of modern databases was performed to find genes with functional association in a particular disease and was ranked according to their local interactions by the STRING database, a database of established gene-gene interactions. The highest ranked genes are predicted to play critical roles in the disease and are defined as “leader” genes. In this way, a far shorter list of relevant genes is created for the purpose of *ad hoc* experimentation.

**Introduction:**

Some methods like genetic linkage and positional cloning are able to identify the genes implicated in simple Mendelian diseases. However, these methods are ineffective for identifying the genes of complex, multifactorial diseases. These diseases are often caused by the contribution of several or numerous genes as well as being expressed in largely varying severity, symptoms, and age or environment of onset (Covani 1974).

While genomic analyses, mainly in microarray technology, seem to be an appropriate technique in solving the genes of complex diseases, this is not the case. This is due in part to the overwhelming amount of data produced by such “omics” techniques, especially when wet lab experimentation plays a vital role in verifying new hypotheses (Bragazzi 165). Moreover, the diseases often encompass a large number of genes, and the simple measurement of relative gene expression is insufficient in understanding the complexity of the network of their interactions. Housekeeping genes as well as genes in other unrelated biological processes significantly contribute to false positives, resulting in redundant and inaccurate analysis.

Freely-accessible databases, such as those hosted by the National Center for Biotechnology Information (NCBI) today do contain the necessary information to make accurate hypotheses about the genes involved in diseases, but without using bioinformatics or computational means, it is almost impossible to correctly extract. The biological database, STRING (Search Tool for the Retrieval of Interacting Genes/Proteins), contains curated data from other interaction databases, enabling the possibility to predict gene networks and expand them (Szklarczyk D363).

The leader genes approach (LGA) makes use of three main algorithms that access these two types of databases: 1) an NCBI data mining algorithm to predict a set of genes with an established functional association to a disease; 2) an expansion algorithm from STRING that is recursively applied to expand the network of interactions and cross-check new genes until convergence is reached; and 3) STRING’s gene interaction scoring algorithm combined with a ranking algorithm to predict the leader genes of the network.For the current implementation, algorithm 2) is functional but is not implemented in the current version of the application.

Applying the LGA approach to well-known diseases can provide broad insights to the molecular basis of those diseases and improve the efficiency of *ad hoc* experiments.

**Methods & Materials:**

Source Code Editor: Sublime Text

Languages:

* R
* Python
* HTML (from R shiny)
* CSS (from R shiny)
* JavaScript (from R shiny and R shinyjs)

Packages/Libraries:

* Python Bio.Entrez
* Python csv
* Python json
* Python time.sleep
* Python urllib3.exceptions.HTTPError
* Python xml.etree.ElementTree
* R DT
* R ggplot2
* R reticulate
* R shiny
* R shinyjs
* R STRINGdb

Databases:

* NCBI – Gene, GTR, MedGen, MeSH, PubMed, OMIM, HomoloGene, ClinVar
* STRING – MINT, HPRD, BIND, DIP, BioGRID, KEGG, Reactome, IntAct, EcoCyc, NCI-Nature Pathway Interaction Database, GO

Front-end: Shiny Web Application

A complete list of all functions in the source code is included in the “Leader Genes Approach – Reference Manual” document.

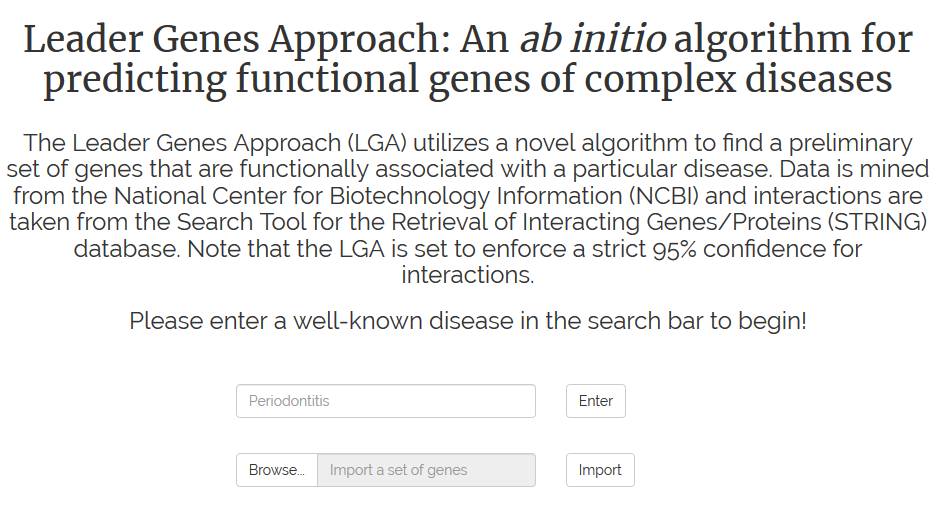
With NCBI Entrez (E-utilities), the search term is built using NCBI MeSH terms, providing a way to check for invalid queries (user input without any MeSH term hits) as well as expanding the search scope. Preliminary genes are mined from NCBI Gene, GTR, and MedGen, and functional associations are embedded within the text of each document. Each preliminary gene is cross-checked for functional association against PubMed by co-occurrence. E-Utilities are used with Python. The final set of cross-checked genes are used as input for STRINGdb.

The interactions between genes are calculated and scored by STRING to create the network via the R STRINGdb package, with a minimum score value of 0.950 to enforce a high degree of confidence. These are derived from seven evidence channels (Szklarczyk D363): 1) Evidence from lab experiment data (biochemical, biophysical, genetic) from the IMEx consortium and BioGRID; 2) Evidence asserted by a human expert curator and imported from pathway databases; 3) Evidence from mentions of protein names in all PubMed abstracts and in other text collections (OMIM, SGD); 4) Evidence from gene expression data that has been normalized, pruned and correlated; 5) Evidence from genome-based prediction, but mostly relevant for Bacteria and Archaea rather than humans; 6) An additional association score is given when a pair of proteins' respective orthologs have fused into a single, protein-coding gene in at least one organism; and 7) Evidence from the phylogenetic distribution of orthologs of all proteins in a given organism. STRING contains a method for expanding the network of genes, and this is used in conjunction with cross-checking in NCBI to recursively expand the network until no new genes are found (convergence is reached). The final scores among all interactions in the network are summed to obtain a ranked list of genes, with the leader genes having the highest scores.

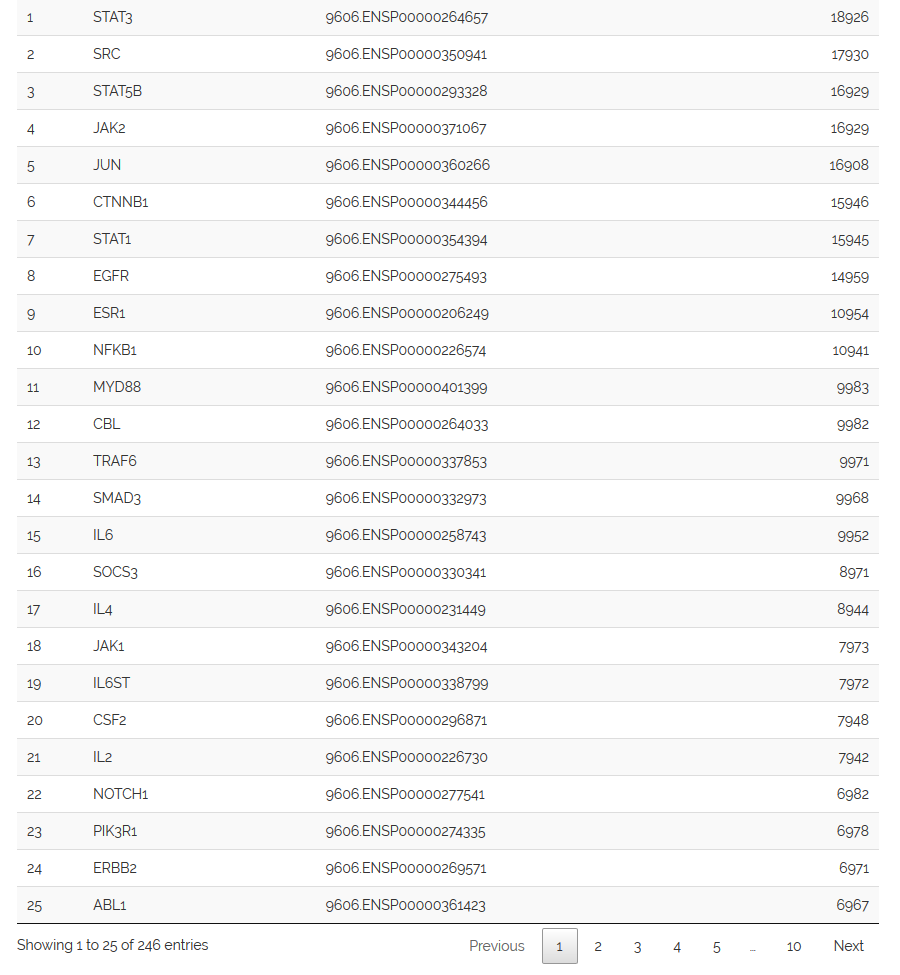
Python is utilized by R via the R Reticulate package.

R Shiny is used to create the web application utilizing HTML, CSS, and JavaScript elements. The title page allows the user to input a search term or import a set of genes to run the algorithms. The results page is split into five different tabs: 1) “Leader Genes” lists the top ten leader genes of the network as well as their global scores; 2) “Network” plots the STRING network and tabulates every single interaction in the network; 3) “Clusters” plots the first five major clusters of the network and the gene interactions within the cluster; 4) “Data” plots the global scores of every gene, including the leader genes and tabulates them; 5) “Export” allows the user to export the list of genes of the STRING network, the table of all gene interactions, and the table of all gene scores.

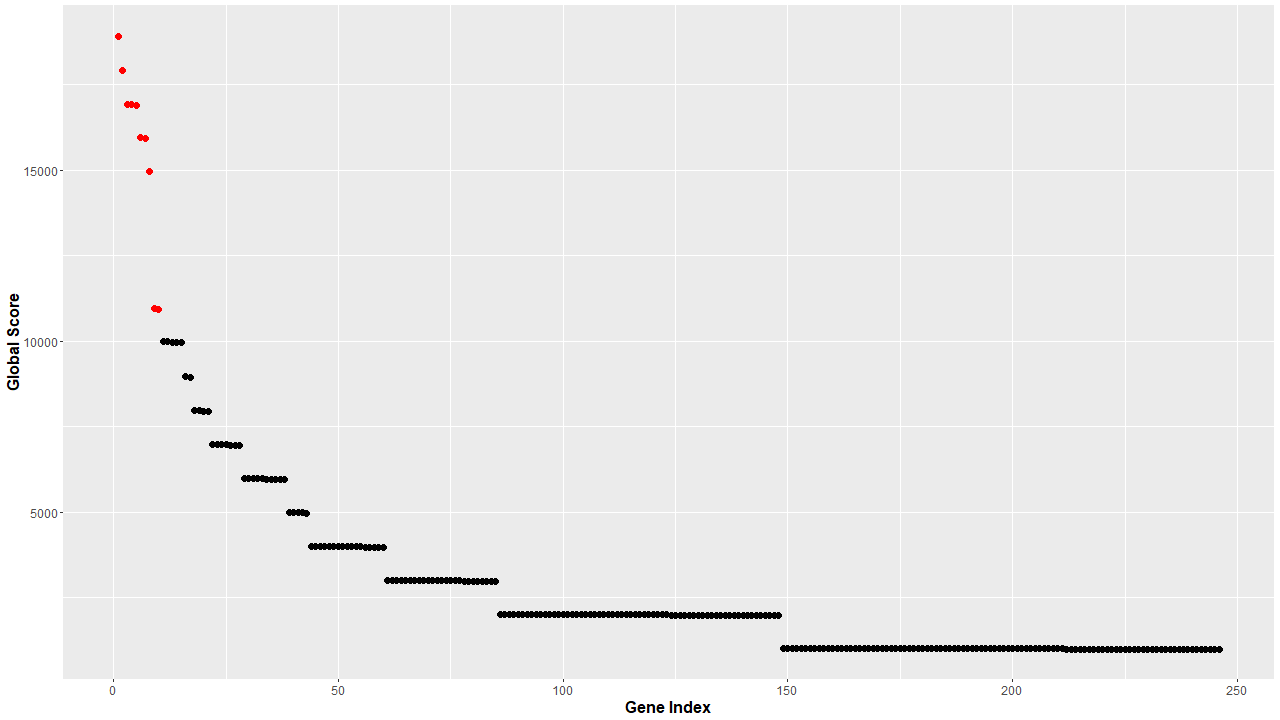
**Results: Run-through of the application with the user input “Periodontitis”**



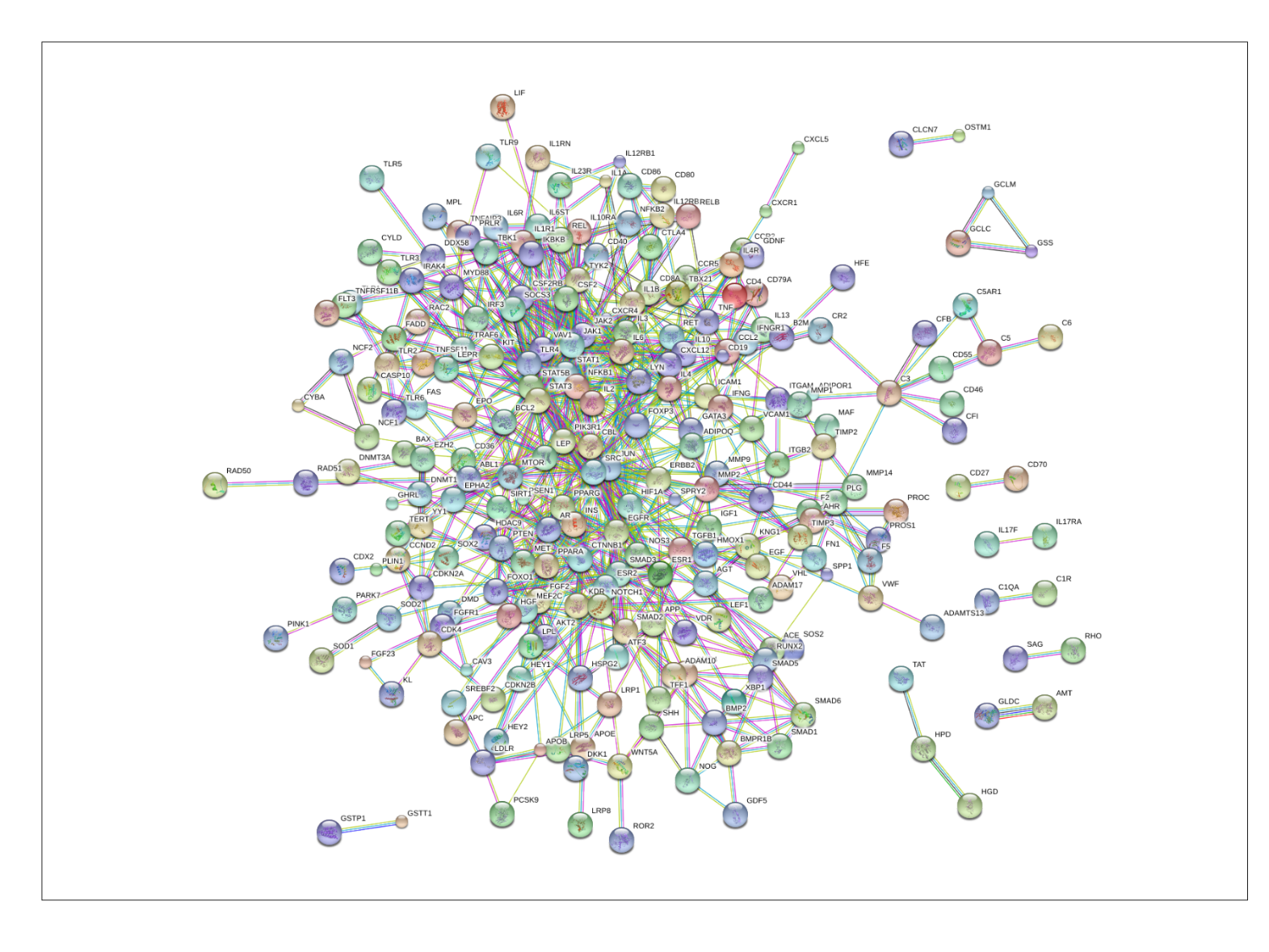
*Figure 1.* The LGA title page gives a brief description of the application, and allows the user two options: 1) enter their own search term to be processed by the application or 2) import a list of genes for the application to process



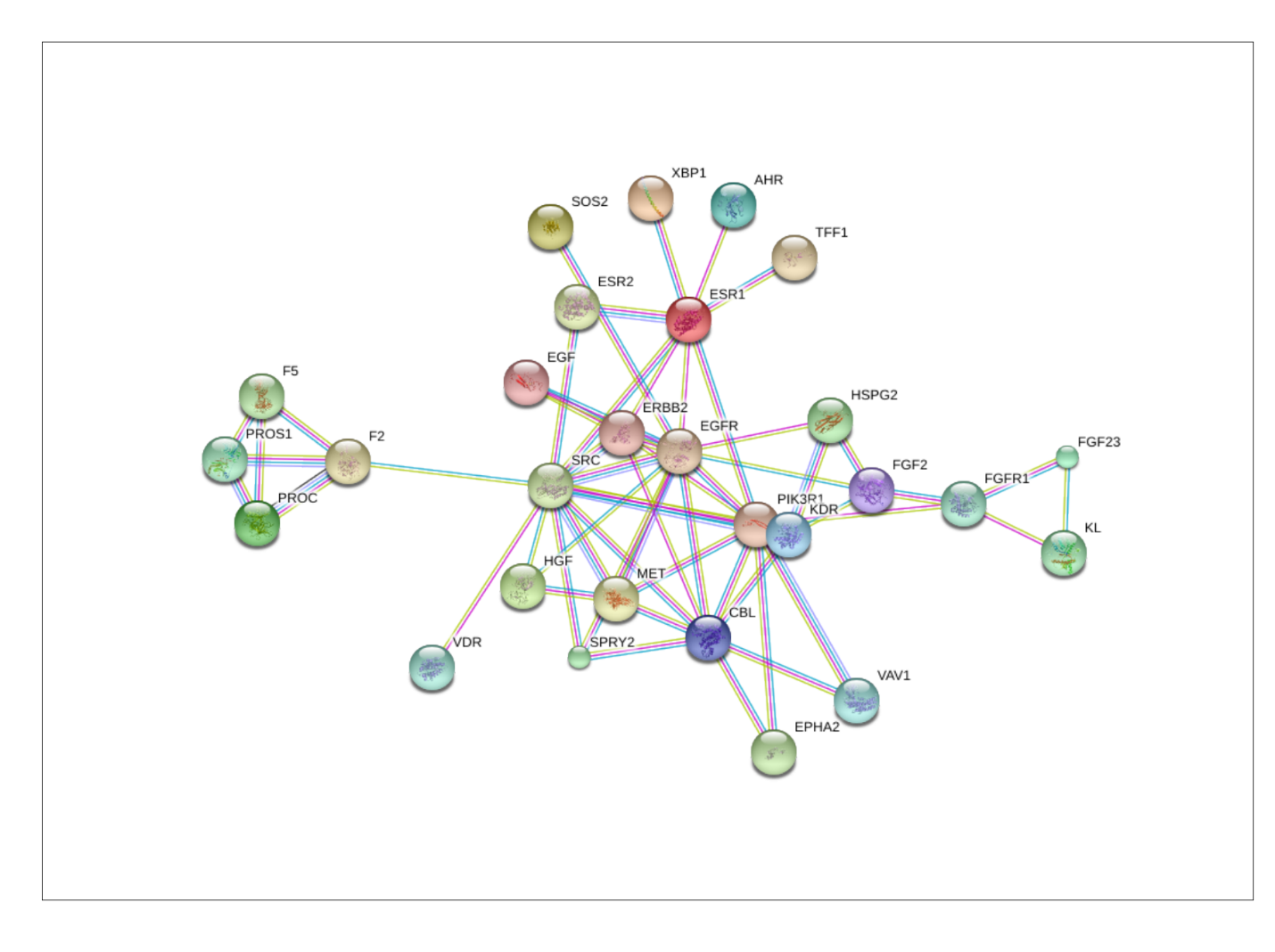
*Figure 2.* The list of all genes in the interactions network that are predicted to have functional association with the disease, currently sorted by Global Score. The application allows sorting by any column.

****

*Figure 3.*Scatter plot of the genes against their Global Score with the leader genes marked in red. The genes are indexed in decreasing Global Score.

****

*Figure 4.* STRING network of the genes and interactions in periodontitis. Each interaction is displayed as a colored line corresponding to one of the seven evidence channels: 1) Experiments – magenta; 2) Database – cyan; 3) Textmining – yellow-green; 4) Co-expression – black; 5) Neighborhood – green; 6) Fusion – red; and 7) Co-occurrence – blue.

****

*Figure 5*. One of the clusters from the periodontitis STRING network.

**Discussion:**

In order to ensure that the user input is valid, the application does an initial search in MeSH to find at least one hit. If there are no hits in MeSH, the term is rendered invalid because there is likely not enough data in NCBI to support an accurate analysis of the disease (or in the case that the user input was complete gibberish). With the data mining, the total runtime of the application has been observed to be between 45 and 90 minutes. The import functionality significantly cuts down this runtime to about 5 minutes or less, allowing the user to instead input their own set of genes as a .csv file. Future updates should allow the processing of data in other formats.

In preliminary mining, the implementation of at least two evidences of functional association (within the database found and then in PubMed literature) ensures a certain degree of confidence that the gene is relevant; it is definitely possible to raise this minimum number and, in the future, can be an additional functionality of the application.

One major algorithm that the program can utilize is an expansion method provided by STRING. This method can expand a given STRING network with its predicted closest neighboring genes. If utilized in conjunction with a cross-checking method that checks each new gene for relevance in the disease in NCBI, then it would allow the application to essentially verify that it did not omit a functional gene. Although a working implementation of this algorithm is in the source code of the application, it was ultimately not utilized in the final version. Unfortunately, the expansion algorithm tends to expand some housekeeping genes into the network which defeats the purpose of the application entirely. Cross-checking with NCBI becomes difficult as many of these housekeeping genes are used as reference genes in some experiments, resulting in false positives. Perhaps a more specific cross-checking method utilizing natural language processing (NLP) would allow the application to screen out the reference genes.

The interaction scores calculated by STRING are quite comprehensive especially when considering the evidence channels. It would be useful for the application to also include the specific sources that led to its assessment, but for now only the types of interactions and their scores are displayed. The functionality of the R STRINGdb package is quite limited at this time. Moreover, some evidence channels such as those in fusion, co-occurrence, co-expression, and neighborhood likely require a lower level of confidence than 0.950 because only a very few number of interactions include these types of scores.

To consider one future application, leader genes of seemingly unrelated diseases can be compared in gene networks such as the one STRING uses to determine any significant connections.

**Conclusion:**

Bioinformatics and computational methods are necessary to extract and present overwhelming amounts of data in a meaningful way. The LGA is a web application capable of predicting functionally associated genes of diseases, mapping their network of interactions, and identifying those leader genes with the strongest interactions all within a relatively short amount of time. Finding leader genes of various diseases can lead to more accurate *ad hoc* experimentation on the implicated genes of complex diseases. Because there is always an influx of new data, the approach will remain relevant to use indefinitely.

**References:**

Bragazzi, Nicola Luigi, and Claudio Nicolini. “A Leader Genes Approach-based Tool for Molecular Genomics: From Gene-ranking to Gene-network Systems Biology and Biotargets Predictions.” *Journal of Computer Science & Systems Biology* 6.3 (2013): 165-176. doi:10.4172/jcsb.1000113

Covani, Ugo, Simone Marconcini, Luca Giacomelli, Victor Sivozhelevov, Antonio Barone, and Claudio Nicolini. “Bioinformatic Prediction of Leader Genes in Human Periodontitis.” *Journal of Periodontology* 79.10 (2012): 1974-1983. doi:10.1902/jop.2008.080062

Donaldson, Ian, Joel Martin, Berry de Bruijin, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D Bader, Katerina Michalickova, Tony Pawson, and Christopher WV Hogue. “PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine.” *BMC Bioinformatics* 4.11 (2003). doi:10.1186/1471-2105-4-11

Sivozhelezov, Victor, Luca Giacomelli, Shubham Tripathi, and Claudio Nicolini. “Gene expression in the cell cycle of human T lymphocytes: I. Predicted Gene and Protein Networks.” *Journal of Cellular Biochemistry* 97.5 (2006): 1137-1150. doi: 10.1002/jcb.20693

Szklarczyk, Damian, John H Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda T Doncheva, Alexander Roth, Peer Bork, Lars J Jensen, and Christian von Mering. “The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible.” *Nucleic Acids Research* 45.D1 (2017): D362-D368. doi: 10.1093/nar/gkw937

von Mering, Christian, Lars J Jensen, Berend Snel, Sean D Hooper, Markus Kripp, Mathilde Foglierini, Nelly Jouffre, Martijn A Huynen, and Peer Bork. “STRING: known and predicted protein-protein associations, integreated and transferred across organisms.” *Nucleic Acids Research* 33.D1 (2004): D433-D437. doi: 10.1093/nar/gki005