**GeneRank, combining Microarray data with prior network knowledge.**

|  |  |
| --- | --- |
| Report Name | Outline Project Specification |
| Author (daf16) | Daniel Luke Fryer |
| Supervisor | Chuan Lu (cul) |
|  |  |
| Module | CS39440 |
| Degree Scheme | GH76 (Artificial Intelligence and Robotics) |
|  |  |
| Date | February 5th 2016 |
| Revision | 1.0 |
| Status | Release |

# Project description

The aim of the GeneRank project is to research, implement and improve on pre-existing algorithms which aim to rank genes within the genome. Genes are ranked by importance within the genome for specific experiments, for instance, if there is a new potential cure for a disease then it is helpful to know which genes are being affected, what that effect means and how strong the effect is. This is often done for a specific organism, e.g. ranking genes within yeast.

Genes can be ranked based on expression data from microarray experiments, whereby genes are, up-regulated, down-regulated or there is no change. Up-regulation means that there has been a fold-increase in expression, e.g. a two fold increase means that the gene doubles production, of for instance RNA. Genes which express more are considered to be of more importance. If a drug changes the expression of a gene then it is to be noted by a higher ranking than unchanged genes for this given experiment.

Ranking based solely on expression data does not build up the perfect picture of importance within the overall network. A gene which does not change its expression levels but does activate its transcription factor will affect genes which are turned on by this transcription factor. This gene then, affected by the drug could be responsible for turning on many other genes which change expression. Based solely on expression data alone, only the genes which are turned on, and therefore are up or down-regulated will be ranked highly. Whereas clearly the gene responsible for turning them on should also have a high ranking.

In order to account for such genes within the ranking system it is important to include the network structure data for the genome. This is achieved by using Gene Ontology (GO) data, one network is created for each section of the GO: Biological Process, Cellular Component and Molecular Function, whereby genes are connected if they share an annotation defined by these networks [1]. The underlying network of genes is combined with expression data from the given experiment to create a ranking system less susceptible to noise which will include the above mentioned gene types in the ranking. As noted within the paper, there are other sources of gene network data and measures which can be used instead of, or on top of GO data.

The original algorithm is based Google’s PageRank algorithm [2]. It uses the same principle that pages, in this case genes, which are connected by many highly ranked pages should be ranked accordingly as the highly ranked pages vote for the page they link to. In this case genes which connect to many highly ranked genes are likely to be genes which control whether or not they are turned on and thus should be ranked highly as well.

The algorithm will be implemented in Python using open sourced elements of the original algorithm. Once implemented and tested the algorithm will be experimented with and extended. It is possible to look at alternative approaches and incorporate these into the algorithm as well as the possible application of such an algorithm to other, similar areas of biology where ranking can prove useful.

# Proposed tasks

The following tasks will be performed on this project:

**Research into the original algorithm along with alternative algorithms**. This will be expanded onto research into expansion of the algorithm along with alternative uses of the algorithm.

**Data collection and manipulation.** Input data must consist of a vector of gene expression levels and gene network information in the form of an adjacency matrix. Therefore data from datasets must be collected and manipulated into a usable format within the algorithm.

**Set up of documentation and development environments**. All documentation and code will be pushed to a private GitHub repository in order to allow for version control as well as backups. The task will be performed using Python and as such the required programs have been installed ready for use.

**Implementation and then expansion of the original algorithm,** which combines the use of expression data with the GO network data to create a sophisticated ranking system.

**Project meetings and journal**. There will be weekly meetings with the supervisor of the project and a journal entry will be made each week. This will be used to guide future activity as well as record past activity and will also be pushed to GitHub.

**Mid-term presentation and final demonstration**. At the mid-term presentation it should be possible to demonstrate the algorithm along with proposed or in-progress extensions and re-use of the algorithm. These will then be shown in full at the final demonstration. Appropriate notes will be made before each demonstration.

# Project deliverables

**Design and plan documentation**. This will detail how the algorithm works, how it was implemented and what tools were used to do this.

**Mid-term demonstration notes**. Notes made from the mid-term demonstration on what was available to be shown and covered and what would be needed for the final demonstration.

**Final algorithm(s) with extension and re-use**. The python code used for the algorithm, any modification and re-use of the algorithm along with all input data and output results will be shown.

**Final report**. This report will cover the entire project outlining what was done, the importance or use of the work along with appendices and discussion about 3rd party tools used for the project, namely for the original algorithm.

**Final demonstration**. A demonstration of how the algorithm works along with the use and importance of the algorithms.

# 4. Initial annotated bibliography

|  |  |  |
| --- | --- | --- |
| [1] |  | R. B. D. J. H. a. D. R. G. Julie L Morrison, “GeneRank: Using search engine technology for the analysis of microarray experiments,” *BMC Bioinformatics,* 2005.  This paper fully works through how the GeneRank algorithm works, with the algorithm, explanations of expression data and GO data as well a synthetic networks used to prove their algorithm. Shows proof of concept, with it being used on a yeast expreiment and improving previous ranking techniques. |
| [2] |  | B. S. M. R. W. T. Page L, “The PageRank citation ranking: bringing order to the web, Technical report,” *Stanford Digital Library Technologies Project,* 1998. |

This paper explains the origional PageRank algorithm used by google search

engine.

A Matlab implementation of the GeneRank algorithm can be found at:

http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-6-233

### In the Electronic supplementary material" section at the end of the paper.