Determining Potential Yeast Longevity Genes via PPI Networks and Microarray Data Clustering Analysis

Bernard Chen[1], Roshan Doolabh[1], Fusheng Tang[2]
1.Department of Computer Science, University of Central Arkansas, Conway AR 72034, USA
2.Department of Biology, University of Arkansas at Little Rock, Little Rock AR 72204, USA

Abstract— **Identification of genes involved in lifespan extension is a pre-requisite for studying aging and age-dependent diseases. So far, very few genes have been identified that relate to longevity. The process of analyzing each single gene one at a time can be a very long and expensive process. It is known that approximately 10% of 6000 yeast genes are lifespan related genes; however, less than 100 genes are identified as longevity genes. The interconnection of multiple genes and the time-dependent protein-protein interactions make researchers use systems biology as a first tool to predict genes potentially involved in aging. In this study, we combined analyses of protein-protein interaction data and microarray data to predict longevity genes. A dataset of all 6000 yeast genes was utilized and a protein-protein interaction ratio was used to narrow the dataset. Next, a hierarchical clustering algorithm was created to group the resulting data. From these clusters, conclusion of 6 highly possible longevity genes was drawn based on the amount of longevity genes in each cluster. Based on our latest information, one of our predicted genes is identified as longevity gene. Wet lab experiments are applied to our predicted genes for supporting the findings.**

***Index Terms*—yeast longevity genes, Clustering, PPI**

I. Introduction

Unearthing new longevity genes in organisms are vital to the study of aging and age-related diseases including cancer and diabetes. It is possible to prolong a cell's lifespan by searching for these longevity genes: genes that are associated with the duration of the cell's life. Once a longevity gene is identified, studies can be initiated to enhance these genes. However, there is a big problem today whereby the gene data size to be analyzed is very large. It could be very costly to search for a longevity gene one-by-one. Analyzing a single gene slide for properties that could sustain longevity can take an extensive amount of time. These gene slides, called microarrays, contain DNA spots with relevant gene information pertinent to geneticists [1]. In order to locate more longevity genes in an efficient manner, a different method must be employed.

Because it can be complex to study human genes, it is more feasible to carry out experiments on a simpler organism. Baker's yeast would provide a solution to such a problem. Yeast is an "easy-to-manipulate" eukaryote cell that would be much easier to use rather than the human equivalent. Furthermore, yeast genes are very similar to human genes. Because it is possible to successfully substitute a human gene for a yeast gene, experimentation on a yeast gene is not only theoretically significant but also applicable to studies on human cells [2]. A yeast cell has less than a fourth of the amount of genes than a human cell making it a simpler, uncomplicated experiment variable [3]. There are around 6000 total yeast genes and only 86 found known longevity genes at the time of our research, making less than 2% of yeast genes known as longevity genes. Based on the current research [4], approximately 10% of the yeast genes are related to lifespan regulation. In order to narrow the complete data down so as to allow for greater success, gene interactions can be analyzed.

Without the aid of advanced computing to facilitate many biological studies, developments cannot be improved considerably. This aid from computing assists to achieve results of a higher magnitude. The discipline of bioinformatics can help biologists and computer scientists solve complex problems in conjunction with one another. There have been several advances in microarray technology which has made it possible to monitor the expression of thousands of gene genomes. One of the main challenges is to analyze and interpret these large datasets in an effective, time efficient way [5]. Exploration of gene expression datasets is always problematic due to its inherent dispersion and missing values [6]. Because of this, dealing with outliers is quite an important issue for analysis methods.

Each cell contains a profuse amount of proteins. Many proteins tend to bind together and form connections to start biological functions. These connections are known as protein-protein interactions [7]. These protein-protein interactions (PPI) are important to the field of bioinformatics because they can show the relationship between genes. Additionally, these interactions provide further information on which genes have more of a potential to be longevity genes. For this paper, our hypothesis is: if a gene has more interactions with a longevity gene, then this gives it a higher probability of being a longevity gene itself. Integrating data that shows the amount of interactions between genes with our computing methods will allow us to limit our research to a smaller dataset. Rather than analyzing all 6000 yeast genes, a much smaller amount of potential genes can utilized.

Our data requires a way to sort granules of information (in our case, single genes) into groups. Each of these groups will contain alike genes. By forming these groups with our data, we will be able to draw strong conclusions on whether a gene has a high potential for being a longevity gene. There are several different ways to group yeast genes. According to Han et al. [8]: "Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are dissimilar to objects in other clusters". Because there is such a large amount of research towards data clustering, there is a large

variety of methods to choose from. After careful consideration of many grouping features, the most prudent algorithm for our data was the hierarchical clustering algorithm.

By combining two methods together, the total yeast-gene dataset can be analyzed much more successfully. Our approach was to combine protein-protein interaction results with a hierarchical clustering algorithm. With this combined methodology, the large dataset was first narrowed significantly by selecting genes that had a high protein interaction with known longevity genes. This was quite a necessary step given that there is a less than 10% chance that any specified gene in the large dataset is a longevity gene. Once the dataset was cut down to include only genes with a high potential to be longevity genes, the new dataset which contains the candidate genes was then grouped using the hierarchical clustering algorithm. This clustering algorithm will help to further associate known longevity genes with generic genes.

II. METHODOLOGY

A. *Applying PPI to the Dataset*

A dataset obtained from the SGD website (http://www.yeastgenome.org/) of each gene's protein-protein interaction with known longevity genes and with all total genes was used to confine the results. The ratio of longevity gene interactions to total gene interactions was then computed. The average of the total ratios was then found as 0.02974 (around 3%). This average was then used to find the standard deviation of the dataset which was found to be 0.052. From this, a threshold of two standard deviations was set in order to narrow the results A total of two standard deviations were chosen because the results gave an optimum dataset that was not too big and not too small and provides the significant statistical meaning. This was done by setting up preliminary groups of genes and finding the best amount of genes which, when grouped, could provide information. Once this process was complete, the list approximately 6000 yeast genes were confined to 197 genes, a significant number reduction. Table 1 shows a small example of the selected genes. Once the dataset was filtered down to 197 potential genes and combined with the 86 known longevity genes, the data was then clustered using the java hierarchical clustering algorithm (described in the next subsection).

Table 1: A section of 15 genes is shown taken from the filtered dataset.

YBL054W	37	5	0.135135135
MSN2	170	23	0.135294118
RPL39	22	3	0.136363636
GLN3	190	26	0.136842105
ATP7	51	7	0.137254902
SFP1	51	7	0.137254902
KAR4	29	4	0.137931034
DAL80	58	8	0.137931034
MPH1	130	18	0.138461538
IDH1	202	28	0.138613861
MEP3	36	5	0.138888889
TDP1	57	8	0.140350877
MMS4	213	30	0.14084507

B. *Clustering Algorithm Selection and Utilization*

After careful consideration of many grouping features, the most prudent algorithms for our data were the hierarchical clustering algorithm and the K-means clustering algorithm. The K-means algorithm is a partitioning method that uses a centroid-based technique. However, the K-means algorithm was found to have drawbacks because the number of clusters must be provided each time the algorithm is run. In addition, noisy values can throw the mean value off thus causing skewed results [8]. The better alternative was found to be the hierarchical clustering algorithm. Unlike the K-means algorithm, no input parameter was required. Additionally, the hierarchical clustering algorithm has been used in the past for microarray analysis [5]. This algorithm has two main types: (1) Agglomerative approach – a bottom-up approach which starts by treating each object as a cluster and forms groups on at a time. When two groups are similar, then they are merged together; (2) Divisive approach – a top-down approach which starts by treating the complete dataset as one whole cluster and then splits this into smaller clusters

With both methods, the user can indicate the desired number of clusters which acts as a termination condition [8]. In this work, the desired method that was chosen was the agglomerative approach. The basic steps for the hierarchical clustering algorithm using an agglomerative approach are as follows:

Repeat

- **Locate two objects (clusters) with the closest distance among all and cluster them together**
- **The values of attributes are changed to reflect the new cluster. This new average is the two old objects averaged together**

Until the hierarchical clustering algorithm generates the desired amount of clusters

The application accepts two parameters upon startup. These inputs are: (1) The number of times to cluster the dataset and (2) The threshold for the chosen number of longevity genes per cluster (expressed as a percent; this parameter will be carefully described in the results section). The dataset that the program used contains two sets of data: the first set of the data contains 197 genes filtered by the PPI_Ratio described in the previous subsection; the second set of the data contains 86 longevity genes. Since the Microarray data [9] that we applied contains three experimental conditions, the algorithm would take three numbers associated with each gene and calculate the distance with other genes. For the purpose of this research, the three experiments are nonessential to this paper and will not be expounded upon.

Several trials were performed to find the optimum number of clusters. With our clustering algorithm, clustering too much would produce large clusters with too much data. These clusters would be impractical because no useful conclusions could be drawn about the relationship between genes. Conversely, clustering too little would produce too many clusters which would again prove hard to extract conclusions about each genes relationship to others. Each time the program clustered was counted an iteration. Figure 1 shows how the total number of clusters (in this paper, we count one cluster when more than two members are grouped together) changes depending on how many time the program grouped the data. It indicates that the number of total clusters increases and then decreases as the number of iterations entered was increased. Also, the information collected

showed that data that fell between 40-60 iterations was useful. Data collected at less than 40 iterations and greater than 60 iterations results in non-beneficial results. For example, after 70 iterations, there were only 6 clusters created each of which contained some clusters which contained greater than 30 genes. This did not provide results that can be used since the clusters were too general. The same principle was found for smaller iterations that were less than 40.

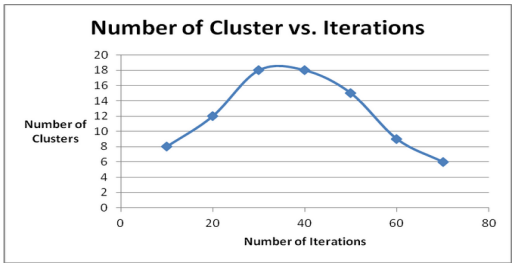

Figure 1– shows the relationship between the numbers of iterations of the clustering algorithm compared to the number of clusters generated

III. Results

The dataset was run by hierarchical clustering at iterations starting at 10 and ending at 60. Figure 2 shows a representation of a cluster result after 40 iterations of clustering.

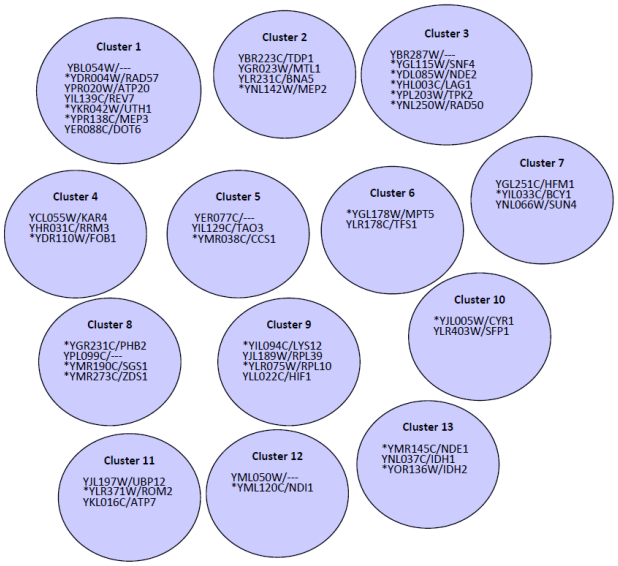

Figure 2 – The clusters with longevity genes after 40 iterations of clustering. (Please note asterisks denote longevity genes)

As shown in the diagram, there were thirteen clusters that were produced. For each cluster, we calculate the percentage of known-longevity genes (genes with * mark in Figure 2) and potential longevity genes. For example, the cluster 3 in figure 2 contains 6 genes, 5 of them are known-longevity genes. Since this cluster contains 83.33% of known-longevity genes, the YBR287W/--- is considered as a highly potential longevity gene. The basic rationale is that the higher the percentage of longevity genes, the greater chance that the potential longevity genes are truly longevity genes since they share common microarray expressions. For instance, if we set the longevity gene threshold to 70%, the example we mention earlier in this section would consider the potential gene as a longevity gene. In this way, we discovered one highly potential longevity gene.

Figures 3-5 show the results based on the number of potential longevity genes found according to different longevity gene threshold. From the results, it can be concluded that the optimum number of iterations was between 40 and 60 iterations. As all three experiments show, there is a peak in potential genes at 50. This bell curve between 40 and 60 proves a good segment of data to use for further analysis. This is so because the optimum number of genes would peak and then decline as clusters contained too many non-longevity genes.

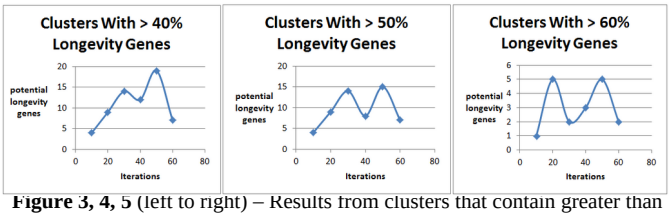

Figure 3, 4, 5 (left to right) – Results from clusters that contain greater than 40%, 50% and 60% longevity genes, respectively.

Although the experiment that counted potential longevity genes in clusters with 60% or greater longevity genes produced a smaller amount of potential genes, the data proved to be the most useful. Genes that are clustered with dominate longevity genes group have a higher potential to be longevity genes. The experiment which uses 60% or greater longevity genes, when compared to others, resulted in more precise results. By narrowing the results so that the potential genes would need to be in a cluster with a large amount of longevity genes, more specific results can be drawn.

By combining the data from the experiment used with 60% or greater longevity genes and by selecting the results within 40 – 60 iterations, a small set of potential longevity genes can be found. The tables 2~4 show the potential longevity genes based on the criteria which narrows the data results. The colors in the table show a good representation of the genes and how important they are. If a cluster of genes contained 60% or greater longevity genes, they were flagged (in red color) as having the most potential to be longevity genes. And while orange and yellow genes weren't as important, they still provide us with several candidate potential longevity genes, only lacking priority.

Table 2

After 40 iterations	Percentage
YBL054W/--- YPR020W/ATP20 YIL139C/REV7 YER088C/DOT6	42%
YBR287W/---	83%
YLR178C/TFS1	50%
YPL099C/AIM43	75%
YJL189W/RPL39 YLL022C/HIF1	50%
YLR403W/SFP1	50%
YML050W/---	50%
YNL037C/IDH1	67%

Table 3

After 50 iterations	Percentage
YBL054W/--- YPR020W/ATP20 YIL139C/REV7 YER088C/DOT6	43%
YBR287W/--- YER077C/--- YIL129C/TAO3	67%
YCL055W/KAR4 YHR031C/RRM3	50%

YJL197W/UBP12 YKL016C/ATP7 YLR403W/SFP1 YNL037C/IDH1	
YDR527W/RBA50 YPL099C/AIM43	67%
YLR178C/TFS1	50%
YJL189W/RPL39 YLL022C/HIF1	50%
YML050W/---	50%

Table 4

After 60 iterations	**Percentage**
YBL054W/--- YPR020W/ATP20 YIL139C/REV7 YER088C/DOT6 YCL055W/KAR4 YHR031C/RRM3 YJL197W/UBP12 YKL016C/ATP7 YLR403W/SFP1 YNL037C/IDH1 YIL009C-A/EST3 YJR098C/---	45%
YDR527W/RBA50 YPL099C/AIM43	75%
YBR073W/RDH54	50%
YPL049C/DIG1 YJL189W/RPL39 YLL022C/HIF1	50%
YBR223C/TDP1 YGR023W/MTL1 YLR231C/BNA5 YPR116W/--- YDR078C/SHU2 YDR428C/--- YGL251C/HFM1 YNL066W/SUN4 YLR178C/TFS1 YBR287W/--- YER077C/--- YIL129C/TAO3	43%
YML050W/---	50%

IV. Conclusion

In this paper, we show an approach to find the candidate yeast longevity genes based on the PPI and Microarray data. We search potential longevity genes based on our proposed PPI_Ratio. Then we put those genes with known longevity genes in the same pool and cluster them with the Microarray experimental results. By using the concept of "Genes that are clustered with dominate longevity genes group have a higher potential to be longevity genes", we are able to further extract high potential longevity genes.

TABLE 5 Highly potential longevity genes found in this research

Potential Longevity Genes
YBR287W/---
YDR527W/RBA50
YPL099C/AIM43
YNL037C/IDH1*
YER077C/---
YIL129C/TAO3

Based on Table 2~4, the most prevalent genes are listed in table 6. Predictions in Table 5 are supported by multiple lines of evidences. First, the identification of YNL037C/IDH1 is consistent with the experimental observation that deletion of IDH1 extends the replicative lifespan [10]. In other words, IDH1 IS proven as a longevity gene. At the time of our experiment, the IDH1 is not yet included in our seed list of longevity genes. The identification of IDH1 by our approaches thus supports the efficiency of our approaches (at least one out of six is already higher than 10%, which may be achieved by just guessing). Second, proteins of 4 (YPL099C/AIM43, IDH1, YER077c, YIL129C/TAO3) out of the 6 predicted longevity genes in Table 5 are localized to mitochondria. Mitochondria generate energy by respiration to support cell growth. Up-regulation of respiration is a longevity mechanism [11]. Third, we compared the DNA sequences of the promoter region (from -1000 nucleotide to the first ATG cocon) of these 6 genes but did not find a common motif, suggesting that these 6 genes are not clustered together merely by the activation of a common transcription factor. Instead, proteins of these genes may interact with each other during anti-aging processes.

Although whether genes other than IDH1 in Table 5 are involved in aging awaits for wet lab validations, previous data about functions and localizations of proteins of these genes strongly support our predictions. Upon completion of this project, it was found that by using computing to aid biological experiments, potential longevity genes were discovered. With this information, several additional trials can now be performed on the results found. Because our results concluded that a small number of yeast genes have a high probability of being longevity genes, biologists can concentrate resources to finding valid empirical results using the final 6 genes that were found in this paper.

References

[1] Stekel, D. (2003). Microarray bioinformatics. Cambridge: Cambridge UP.

[2] Pines, M. (2011). Robot that tracks all the genes in a cell. Retrieved from http://www.hhmi.org/genesweshare/a110.html

[3] KOLATA, G. (1996, April 26). Map of yeast genes promises clues to diseases of humans. Retrieved from http://www.nytimes.com/1996/04/25/us/map-of-yeast-genes-promises-clues-to-diseases-of-humans.html

[4] Kaeberlein M, Powers RW 3rd, Steffen KK, Westman EA, Hu D, Dang N, Kerr EO, Kirkland KT, Fields S, Kennedy BK. 2005. Regulation of yeast replicative life span by TOR and Sch9 in response to nutrients. Science. 310(5751):1193-6.

[5] Chen, Bernard, Phang C. Tai, R. Harrison, and Yi Pan. "Novel Hybrid Hierarchical-K-Means Clustering Method (H-K Means) for Microarray Analysis." CSBW '05 Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (2005): 105-08.

[6] Quackenbush, J. (2001) Computation analysis of microarray data. Nat. Rev Genet., 2, 418-427.

[7] De Las Rivas J, Fontanillo C (2010) Protein–Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks. PLoS Comput Biol 6(6): e1000807. doi:10.1371/journal.pcbi.1000807

[8] Han, J., Kamber, M., Pei, J., Pei, J., & et al, (2012). Data mining, concepts and techniques. (3rd ed.). Waltham, Ma: Morgan Kaufmann.

[9] Cheng C, Fabrizio P, Ge H, Longo VD, Li LM. 2007. Inference of transcription modification in long-live yeast strains from their expression profiles. BMC Genomics. 8:219.

[10] Delaney JR, Sutphin GL, Dulken B, Sim S, Kim JR, Robison B, et al, Sir2 deletion prevents lifespan extension in 32 long-lived mutants. Aging Cell. 2011 Dec;10(6):1089-91.

[11] Lin SJ, Kaeberlein M, Andalis AA, Sturtz LA, Defossez PA, Culotta VC, Fink GR, Guarente L. Calorie restriction extends Saccharomyces cerevisiae lifespan by increasing respiration. Nature. 2002 Jul 18;418(6895):344-8.