Rice University BioSciences Department

Computational Analysis of Biological Datasets

Final Project Paper

Computational Identification of Common ESRE Motifs

Adam Howard

In cooperation with the Kirienko Lab

December 7, 2017

Introduction

Caenorhabditis elegans, a bacterivores nematode, is a well characterized model organism supported by a diverse array of genetic tools. One advantage of studying gene pathways in these small animals over other types of model organisms, such as yeasts and plants, is their strong sequence homology with human genes as well as similar expression mechanisms to other animals. One of the commonalities between *C. elegans* and humans is the high sequence homology between our innate immune genes (6). All of these factors highlight *C. elegans* potential as a model for drug-induced immune stimulation.

This project, conducted in cooperation with Dr. Natasha Kirienko's lab, focuses on discovery of novel small molecules that can serve as immunostimulants in patients with compromised immune systems, such as individuals with cystic fibrosis or cancer. A large library of potential drugs was used to determine if the lifespan of *C. elegans* could be extended in response to a known pathogen, *Pseudomonas aeruginosa.* From this study, six small molecules (referred to as LK molecules) were ultimately determined to potentially upregulate the immune system. Microarray analysis revealed a large list of genes that were impacted in response to these small LK molecules. By analyzing their promotor regions for common elements, we may be able to determine if there are any pathways that are common between these genes, as well as possible sites for transcription factor binding. This study is set to engage in this motif discovery using a simple string finding approach. While several issues were encountered along the way which caused a reduction in sample size, overall it was possible to find a specific given motif with this program. More elegant solutions may exist, but this project allowed me to engage with the complex problem of motif finding first hand.

Methods

Prior to my joining the project, several microarray analyses were conduct to study the gene expression changes in *C. elegans* in response to each of our small molecules. For this project, only genes that were upregulated in response to the small molecules were considered. Additional microarray data were included from published gene changes in response to other chemical stressors to test for similarities between the new small molecules and known stress pathways. For each gene, the fold changes were normalized across the conditions by the formula:

Equation 1. $$\frac{x_i - mean(x)}{stdev(x)}$$

Where x_i represents the gene expression fold change for a compound and x is the set of all possible expression values for that gene across all of the compounds. Data were stored in an Excel file.

Using the *xlsread* command in MATLAB, I was able to read the Excel file into two array variables; one for the text and one for the numerical expression values. I then called the *kmeans* function, which performs K-means clustering based on my expression data. *kmeans* accepts the expression data and the number of clusters saved in a variable as its two minimum arguments. A loop that draws subplots of the expression data from each cluster was used to evaluate the best number of clusters by inspection of their graphs. The key distinguishing factors between clusters were the mean of the cluster, as well as the overall shape of the cluster's graph. Cluster IDs taken as the output of *kmeans* were then applied to the text variable and sorted into descending cluster order for easier manipulation of the data.

A structure was designed, called *Kgroups*, which will store all gene information in an organized way (Fig. 1). The sorted data from the *kmeans* clustering was stored as each field of

the structure along with several additional fields to hold information related to the sequence of the genes and their promotor regions. The structure fields were filled using a vector-wise approach.

```
%Make a structure for my gene informaiton.
field1 = 'id';                  value1 = {sortedtxt{:,1}};
field2 = 'geneName';            value2 = {sortedtxt{:,2}};
field3 = 'GenbankID';           value3 = {sortedtxt{:,3}};
field4 = 'GOterms';             value4 = {sortedtxt{:,4}};
field5 = 'GeneSequence';        value5 = cell([1,length(sortedtxt)]);
field6 = 'Promotor';            value6 = cell([1,length(sortedtxt)]);
field7 = 'PromotorComplement'; value7 = cell([1,length(sortedtxt)]);
Kgroup = struct(field1,value1,field2,value2,field3,...
    value3,field4,value4,field5,value5,field6,value6,field7,value7);
```

Figure 1. Creation of the structure fields used to hold sorted cluster information as well as the sequence of the gene and promotor regions. Gene ontologies were saved, but not explored in this analysis.

Following the population of the *Kgroups* fields, the *getgenbank* function was utilized to pull up the DNA sequences for each of the genes. These sequences were used to locate the promotor region of the gene designated by 1,000 base pairs ahead of the gene. In the event that the promotor region is less than the 1,000 base pairs, the smaller region that is present will be recorded. Genes were searched for using the *strfind* function against the entire genome which was saved as a string array. The total genome was read in to MATLAB from a .fasta file sourced from WormBase ParaSite website (1-3). Both the forward and reverse DNA strands for the entire genome were used. Initial search yielded very few promotor regions, so the code was modified to search for only the smallest number of nucleotides needed to be unique in a genome, as defined by the following formula:

Equation 2. $$Minimum\ Unique\ Bases = \log_4(Genome\ Size).$$

This formula was selected based on its similarity to the problem of unique primer design in PCR and cloning. Promotor regions were searched using *strfind* for the Ethanol Stress Response element, or ESRE, (tctgcgtctct), and looped such that it searched every promotor stored in

Kgroups. Finally, number of positive ESRE hits were counted and compared to the total number of ESRE hits in the genome using a hypergeometric probability test.

Results

My original gene list generated from my microarray data analysis in Excel included 939 genes. Only genes that had assigned GenBank IDs could be used at this time to pass to the *getgenbank* command. Prior to importing my genes, all genes that did not have a GenBank ID associated were removed, leaving 697 genes in my list to import into MATLAB.

To begin selection of my K-means clustering, a hierarchical clustering performed previously in MATLAB was used to first estimate the number of clusters. By inspection, approximately 10-17 clusters emerged. Each were tested, finally resulting in 15 clusters as the maximum number of clusters that could be used without artificially subdividing groups (Fig 2).

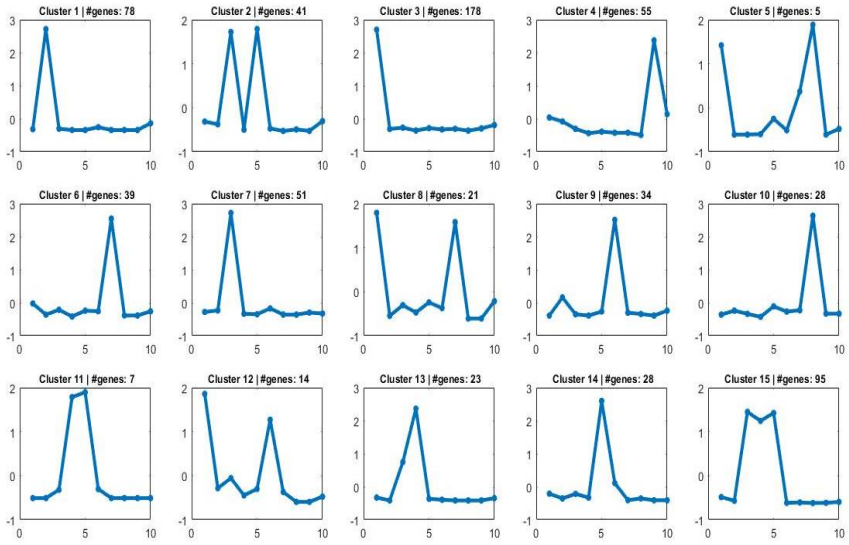

Figure 2. Graphs of the means of each cluster for the 15-cluster analysis. The shape of each subplot in the above figure appears to be unique, indicating that it is unlikely that any of these clusters are artificially subdivided.

We also observed a fairly wide distribution in the number of samples that sorted into each cluster (Fig.3). While this distribution somewhat fits the expected trend that the majority of genes would fall into a non-specific stress response pathway, further analysis is still required to verify this trend.

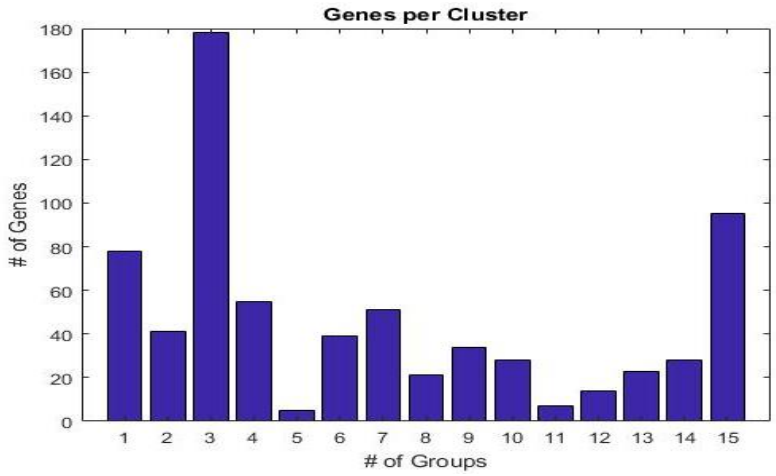

Figure 3. Number of Genes sorted into each group. The distribution of genes sorted into each cluster varies greatly (Mean = 44, Range 173). We expect to see the majority of genes associated with general stress responses, but further analysis is required to test this.

The published *C. elegans* genome size is 100,258,171 base pairs (2,3). Using Equation 2, we find that the base length required to be unique in a genome of this size is at least 14 base pairs. The string finding result returned 500 promotor regions, leaving 197 unidentified promotors. The limited return of promotor regions could be due to small point differences between the 14-base search and the published genome. Out of the remaining 500 promotor regions identified, the ESRE search yielded a total of 11 genes with the motif (Fig 4).

A search of the entire genome for ESRE found 1,433 positive regions. These values were used to conduct a hypergeometric probability test. This test determines how likely a sample subset is to be drawn randomly from a data set that has a certain number of possible

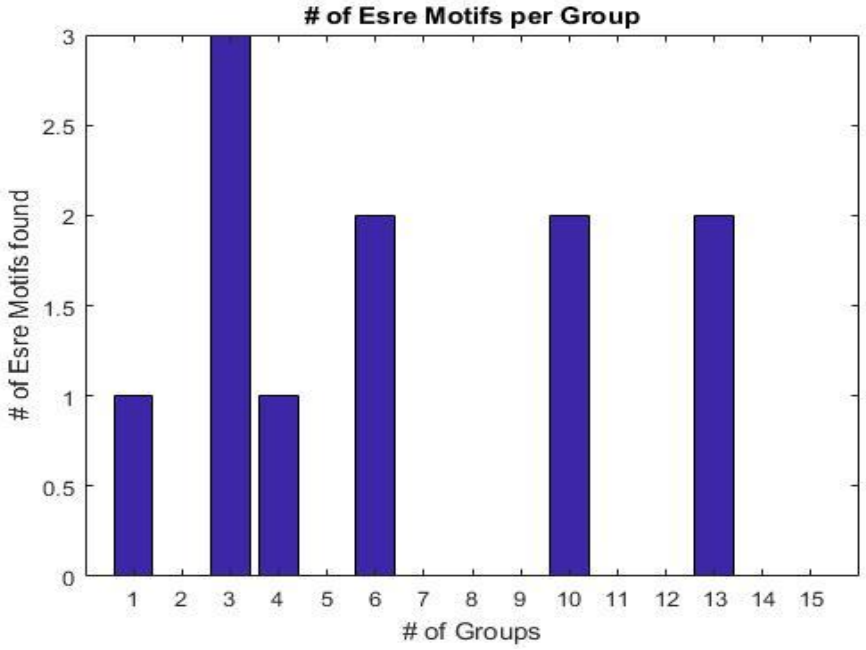

Figure 4. Number of ESRE motifs discovered in each cluster group. The largest set of genes discovered was 3 taken from the largest gene cluster with 178 genes total. The total number of ESRE found is 11 in my gene set out of 1433 in the entire genome.

successes in it. The p-value is greater than 0.992, showing that selecting this set of genes is not statistically different from a randomly selected set of genes. It is important to note that these results were only for the 500 promotor regions identified using string finding out of the 939 genes we began with from the microarray analysis. This reduction in sample size has potential to skew the data set because we cannot assume that the genes removed from the data set were done so randomly and with no effect on the population distribution.

Discussion

While the string finding approach to genome wide motif finding is able to find a known motif, we lost a large portion (28%) during the analysis. The most likely factor disrupting the promotor search are nucleotide discrepancies between the published gene sequence and published genome. *strfind* relies on an exact match in texts. Without a perfect alignment, no hits will be recorded. One solution to make the current version of the code work would rely on searching the genome using increasingly shorter regions from the start of the gene. This will increase the chance of finding a matching promotor region at the risk of lowering specificity to our gene of interest. While this method would not correct all the short comings with this algorithm, it would fix the immediate issue of missing promotors.

A more complete, and perhaps elegant, approach would be to use a local blast search against the genome. A blast algorithm breaks down the larger "phrase" you wish to search, in this case a target gene, into discrete words that are searched for in the larger "dictionary," the genome. As matches are found, longer and longer words are compared, extending from the initial matched nodes. Possible matched regions are scored based on sequence alignment, providing a ranking system characterizing not only the matched region, but how well it matches. This would correct for small discrepancies between the published sequences of the genes and the genome

used, though returning the index of the matched region in the genome is still something that would not be returned as readily from this type of analysis.

While I have explored the implementation of a motif finding system with my project, this is an extensively researched topic among computational biologists. A surprising number of motif finding algorithms have already been created and many are publicly available online. An example of one such finder is *MotifCatcher,* designed by Phillip Seitzer, *et al*. (4). While I have not been able to make direct comparisons between the efficacy of *MotifCatcher* and either my current or proposed motif finding algorithm, a clear advantage to using *MotifCatcher* is the well-developed GUI which is essential to making code in MATLAB accessible to people who have less of a programming background. While many tools are constantly improving, allowing us to find these motifs, it is equally important to determining the biological significance of our findings. Narasimhan, *et al*. (5) discovered a number of false positive results using motif finding approaches. This all points to the need to not only further develop our computational tools, but also continue to verify the biological relevance of our findings.

References

(1) C. elegans Sequencing Consortium. Genome sequence of the nematode C. elegans: a platform for investigating biology. Science. 1998 Dec 11;282(5396):2012-8. Review. Erratum in: Science 1999 Jan 1;283(5398):35. Science 1999 Mar 26;283(5410):2103. Science 1999 Sep 3;285(5433):1493. PubMed PMID: 9851916.

(2) Howe KL, Bolt BJ, Shafie M, Kersey P, Berriman M. WormBase ParaSite - a comprehensive resource for helminth genomics. Mol Biochem Parasitol. 2017 Jul;215:2-10. doi: 10.1016/j.molbiopara.2016.11.005. Epub 2016 Nov 27. PubMed PMID: 27899279; PubMed Central PMCID: PMC5486357.

(3) Howe KL, Bolt BJ, Cain S, Chan J, Chen WJ, Davis P, Done J, Down T, Gao S, Grove C, Harris TW, Kishore R, Lee R, Lomax J, Li Y, Muller HM, Nakamura C, Nuin P, Paulini M, Raciti D, Schindelman G, Stanley E, Tuli MA, Van Auken K, Wang D, Wang X, Williams G, Wright A, Yook K, Berriman M, Kersey P, Schedl T, Stein L, Sternberg PW. WormBase 2016: expanding to enable helminth genomic research. Nucleic Acids Res. 2016 Jan 4;44(D1):D774-

80. doi: 10.1093/nar/gkv1217. Epub 2015 Nov 17. PubMed PMID: 26578572; PubMed Central PMCID: PMC4702863.

(4) Seitzer P, Wilbanks EG, Larsen DJ, Facciotti MT. A Monte Carlo-based framework enhances the discovery and interpretation of regulatory sequence motifs. BMC Bioinformatics. 2012 Nov 27;13:317. doi: 10.1186/1471-2105-13-317. PubMed PMID: 23181585; PubMed Central PMCID: PMC3542263.

(5) Narasimhan K, Lambert SA, Yang AW, Riddell J, Mnaimneh S, Zheng H, Albu M, Najafabadi HS, Reece-Hoyes JS, Fuxman Bass JI, Walhout AJ, Weirauch MT, Hughes TR. Mapping and analysis of Caenorhabditis elegans transcription factor sequence specificities. Elife. 2015 Apr 23;4. doi: 10.7554/eLife.06967. PubMed PMID: 25905672; PubMed Central PMCID: PMC4434323.

(6) Gravato-Nobre MJ, Hodgkin J. Caenorhabditis elegans as a model for innate immunity to pathogens. Cell Microbiol. 2005 Jun;7(6):741-51. Review. PubMed PMID: 15888078.