JASPAR information is part of a new feature of GRNsight that increases the application's ability to interpret the results of gene profiles

Antonio Porras - Department of Biology

Edward B. Bachoura - Department of Computer Science

Quinn Lanners - Department of Mathematics

Simon Wroblewski - Department of Computer Science

Biological Databases CMSI/BIOL 367-01

December 15, 2017

Introduction

Saccharomyces cerevisiae, commonly known as yeast, is a eukaryotic model organism for the study of gene expression (DeRisi et al., 1997). Findings from research of *Saccharomyces cerevisiae* can be extrapolated upon to make inferences about other eukaryotic organisms, including *Homo sapiens*. Yeast has several characteristics that make it a model organism for gene expression research. Among these favorable characteristics include the fact that it is quite easy to manipulate in a lab setting and the availability of its genome sequence (Yourgenome, 2016).

One of the atmospheric stressers that has been tested on yeast is the organism's response to cold shock (Becerra et al., 2003). From analyzing the yeast genome response to cold shock, profiles of genes that are affected by cold shock can be identified. From these profiles, which contain genes that perform a similar function, the overall effect of cold shock on the expression of the genome can be determined. And thanks to the fact that *Saccharomyces cerevisiae* is a model organism, these results can then often be expanded to the human genome (University of Michigan, 2009).

Previous research on yeast and cold shock suggests that temperature change does indeed have a significant impact on the expression of the genome (Becerra et al., 2003). While the number of genes affected by severe cold shock conditions has been found to be less than the number of genes affected by severe heat shock conditions, there are still over 100 genes that have been found to have a significant change in expression when subjected to a temperature change of 30°C to 4°C. However, many of the genes that are upregulated by extreme cold shock conditions are also upregulated by severe heat shock conditions. This suggests that these gene clusters share regulatory signals that are not just for cold shock response, but a larger environmental stress response in general (Becerra et al., 2003).

While progress has been made towards determining the effect of cold shock on the genome expression of yeast, there is still a lack of knowledge regarding which transcription factors in yeast belong in the gene regulatory network that controls the cold shock response of the organism. The deficits of past technology have made the determination of this quite difficult. Programs like GRNmap can conduct mathematical models from microarray results to determine this type of information (GRNsight). However, the results from a GRNmap model, which are returned in the form of a tabular spreadsheet, are incredibly difficult to comprehend and draw conclusion from (GRNsight).

Thankfully, new technology in the form of a program called GRNsight, allows for the visualization of this outputted tabulated data of gene regulatory networks as a diagram that shows the relationships between genes and regulatory relationships as nodes and edges (Dahlquist et al., 2016). Using this software, results from microarray data, and specifically results of different profiles affected by cold shock in yeast, can be formatted and uploaded to create these graphs. These diagrams can then be used to make more confident interpretations regarding which transcription factors belong in the gene regulatory network and what the relationship is between all of the transcription factors in any given network (Dahlquist et al., 2016).

This research project seeked to understand the gene regulation network of one of the most affected gene profiles (profile 22) from the effects of cold shock on the dZAP1 strain of yeast.

Through the statistical analysis of collected microarray data from Dr. Dahlquist's lab on the genomic expression of this strain of yeast at varying time intervals after being subjected to cold shock, the relationship between the most important transcription factors in gene profile 22 were determined. GRNsight was then used to visualize these relationships. Additionally, this project seeked to add to the functionality of GRNsight, which already creates a visualization of the gene regulatory network as nodes and edges (Dahlquist et al., 2016), by adding a new feature in the form of a right-click page that shows general information specific to each gene in the network. This particular team retrieved and returned important information regarding transcription factors from the JASPAR database (JASPAR, 2017), which was then included on the newly created gene information page that was part of a collaborative effort of four teams.

Materials/Methods

Quality Assurance/Project Manager:

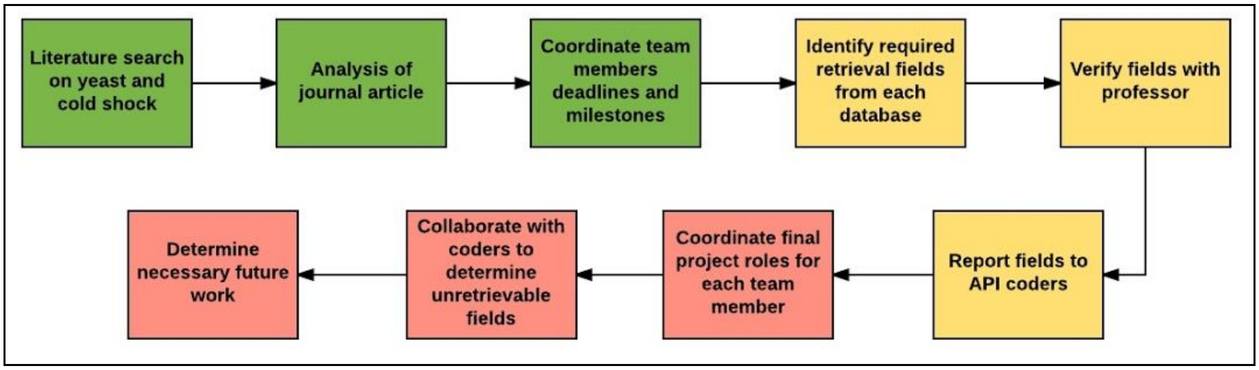

Literature search on yeast and cold shock
Analysis of journal article
Coordinate team members deadlines and milestones
Identify required retrieval fields from each database
Verify fields with professor
Report fields to API coders
Coordinate final project roles for each team member
Collaborate with coders to determine unretrievable fields
Determine necessary future work

Figure A: Flow Chart of tasks/milestones performed by the quality assurance/project manager team member of the JASPAR the Friendly Ghost team.

A number of steps were carried out by the quality assurance/project manager team member. As a starting point, in order to gain background knowledge on the field, the quality assurance team member partnered with the data analyst to perform a literature search for scholarly articles on yeast and cold shock (QLanners_Week_11). After compiling a list of articles, one article was chosen to be reviewed in depth and summarized in the form of a journal club presentation to the class (QLanners_Week_12). Following the conclusion of the background research, the quality assurance team member began to facilitate the necessary actions for the team by determining deadlines and milestones for each member. These deadlines and milestones were formatted into a group calendar (JASPAR_the_Friendly_Ghost). Once these milestones had been established, the quality assurance team member began work on the project by visiting and familiarizing himself with each of the five databases used in this study; which where Ensembl, UniProt, NCBI, SGD, and JASPAR. Once the information included on each database was understood, the quality assurance team member determined the necessary fields to be pulled from each database, compiling a final list of desired fields about each gene and from which database each of these fields should be pulled (QLanners_Week_14). These results were then verified with the professor, adding any additionally requested fields, and then this list was shared with the API coders.

Once the necessary fields had been determined, and the coders had completed their code to retrieve these fields, the quality assurance team member began work on facilitating the final deliverables (QLanners_Week_15). The roles for each team member were determined, and added to the team calendar. Along with determining these final milestones, the quality assurance team member checked-in with the coders from both API teams to identify which fields were unable to be retrieved. Taking into consideration these unretrievable fields, the quality assurance team member then determined the necessary future work.

Data Analyst:

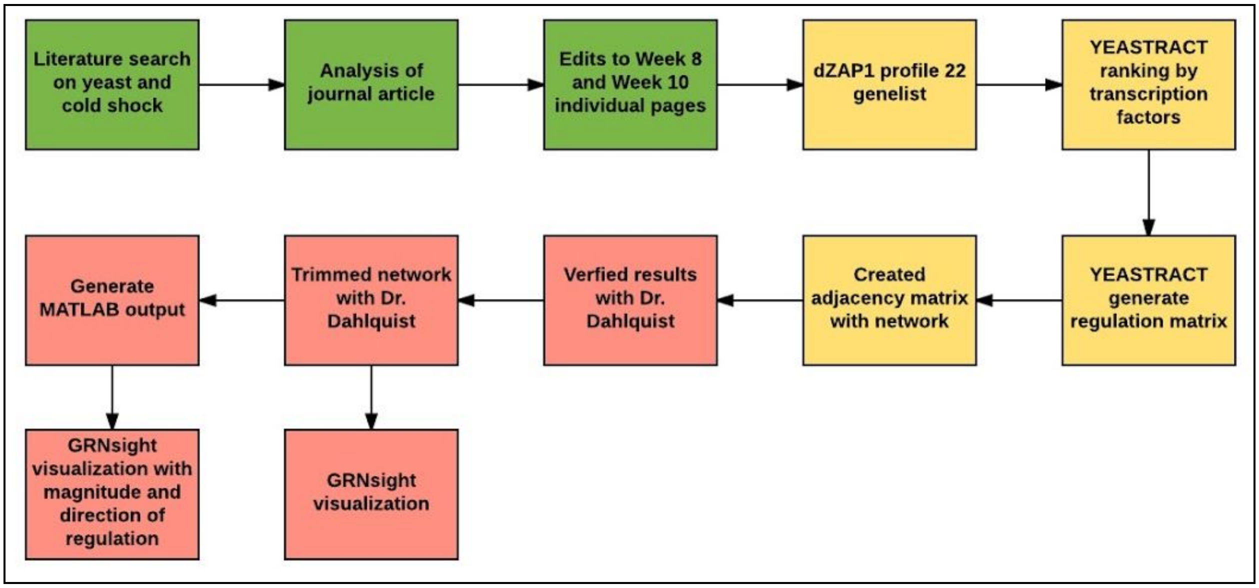

Literature search on yeast and cold shock
Analysis of journal article
Edits to Week 8 and Week 10 individual pages
dZAP1 profile 22 genelist
YEASTRACT ranking by transcription factors
YEASTRACT generate regulation matrix
Created adjacency matrix with network
Verfied results with Dr. Dahlquist
Trimmed network with Dr. Dahlquist
Generate MATLAB output
GRNsight visualization with magnitude and direction of regulation
GRNsight visualization

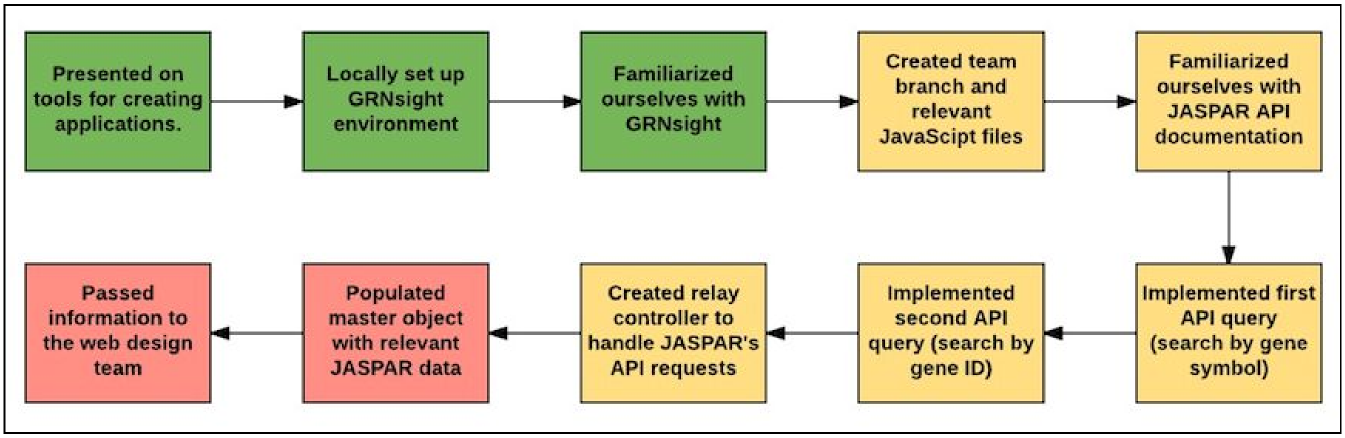
Figure B. Flow chart generated from key tasks and processes in interpretation and visualization of dZAP1 gene expression (Aporras1_Week_15).

The data analyst was assigned the role of performing high level statistical analysis on microarray data of the dZAP1 strain of *saccharomyces cerevisiae*. Before starting analysis, the data analyst worked alongside the quality assurance team member to complete a general search on literature regarding yeast and cold shock (Aporras1_Week_11). Subsequently, one article was selected to analyze in depth and create a presentation in order to present the background, methods, results, and conclusions of the article (Aporras1_Week_12). Prior to beginning analysis with the dZAP1 strain genelist, the ANOVA test was performed to calculate if there was any significant change in the genes different from zero at any time point throughout the collection of microarray data (Aporras1_Week_8). Furthermore, the calculation of the Bonferroni corrected p-value was performed as a more stringent calculation compared to the Benjamini-Hochberg corrected p-value in order to account for the multiple testing problem which can also be found in the results section (Aporras1_Week_8). Once these statistical methods were performed using the microarray data of dZAP1, STEM (Short Term Expression series Miner) was used to cluster and generate profiles of gene expression through the different time points of cold shock and recovery (Aporras1_Week_10). STEM generated seven significant profiles with significant p-values and varying amount of genes within the individual profiles whose GOlists and genelists were saved for future analysis and interpretation. Ultimately, Profile 22 of the dZAP1 strain was selected because it possessed a large cluster of genes, totaling 252, it had a significant p-value of 2.0 E-157, and it displayed no significant change in expression until the recovery phase of the experiment after time point 60 minutes (Aporras1_Week_10). The GOlist provided allowed for analysis of cell processes being altered during the cold shock and recovery phases and the respective genes associated.

Prior to the analysis of the specific profile, edits were made to both Aporras1_Week_8 and Aporras1_Week_10 pages from feedback received from the professor (Aporras1_Week_14). Profile 22 of dZAP1 strain was selected for further interpretation and the genelist was inputted into YEASTRACT's "Rank by TF" to generate a list of transcription factors regulating the genes within the cluster. Once the list of transcription factors was generated, 20 significant transcription factors were selected from the list to create a gene regulatory network in addition to two transcription factors HAP4 and GLN3 (Aporras1_Week_14). YEASTRACT was used to generate a regulation matrix from the selected transcription factors with the filter of "Only DNA binding evidence" (Aporras1_Week_14). This regulation matrix was then formatted prior to being inputted into GRNsight to generate a visualization of the regulatory network of the selected transcription factors and the results were verified by the professor (Aporras1_Week_14).

Due to the large size of the regulatory network, it was trimmed down to a total of 15 transcription factors by the professor and regulatory network visualization was redone to only include 15 transcription factors (Aporras1_Week_15). MATLAB was then used to generate an output which allowed GRNsight to display color signifying the magnitude and direction of regulation. (Aporras1_Week_15).

Coders:



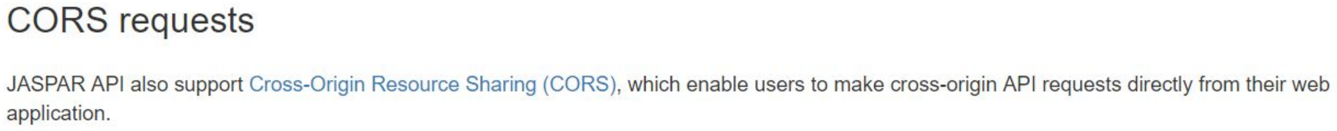
Presented on tools for creating applications.
Locally set up GRNsight environment
Familiarized ourselves with GRNsight
Created team branch and relevant JavaScipt files
Familiarized ourselves with JASPAR API documentation
Implemented first API query (search by gene symbol)
Implemented second API query (search by gene ID)
Created relay controller to handle JASPAR's API requests
Populated master object with relevant JASPAR data
Passed information to the web design team

Figure C: Coders wrote code to retrieve data from JASPAR and delivered results to the design team, accomplishing the above along the way.

There were a number of steps that the coders team members completed throughout this assignment. Before the coders moved their focus to the bulk of the GRNsight part of this project, they were asked to gain some background knowledge on the field of web development. They read “Chapter 6: How Are Apps Made?” in Paul Ford’s *What is Code?* which is an article from 2015 that covers most of the bases of the software development industry. They were asked to thoroughly read this chapter and then present to their class what it entailed. This chapter was specific to all of the tools that the coders used in development, highlighting some important tools at the time. After covering all of the background knowledge, the coders then began the introduction phase to the new project. Often when being assigned to a new project, software engineers are given an introduction to the code base so that they may understand how everything functions within the project before they begin their work. This project was no different, as the professor gave a quick introduction along with a manual which illustrated the relevant folders within the code bank that would be used for this feature. After JASPAR's fellow team members forked GRNsight, the JASPAR team start by setting up a local version of GRNsight and making sure that everything was properly functioning. After this step, they began to acquaint themselves with GRNsight and understand their job in the implementation of this new feature. At this point, as as displayed in Figure C, the coders moved from the immersion phase (green) and into the intermediate phase (yellow).

The intermediate phase was where most of the progress was made towards finishing their part of the feature. They started by creating the branch that would be solely for JASPAR team progress, as well as the new files that would contain their soon-to-be-written code. At this point the coders were completely set up with respect to GRNsight and were ready to learn about the JASPAR database and how it worked. The majority of this part was spent reading the JASPAR database documentation which clearly states how the database API calls work; specifically URL, return values, and headers. JASPAR was effective in laying out their different API calls and explaining what each call did. This made it much easier for the coders to decipher which of the calls was best for retrieving the information that they needed (JASPAR API).

After studying JASPAR, and deciding which calls were to be use, the coders then moved on to implementing the first of the API calls. The first call used the inputted gene symbol to search the JASPAR database and retrieve a list of search results, from which they retrieved the matrix ID from the first result. The second call retrieves the object of gene information associated with that gene symbol. It used the matrix ID that was retrieved from the first call and returned the object with all of the information that JASPAR has for this gene, if any at all. At this point, as they were testing their code, they were getting something called an “Access-Control-Allow-Origin” error which is usually caused by an APIs lack of support for Cross-Origin Resource Sharing(CORS), but when they went and checked the websites overview page, they found the following:



CORS requests

JASPAR API also support Cross-Origin Resource Sharing (CORS), which enable users to make cross-origin API requests directly from their web application.

Figure D. Screenshot of JASPAR’s support of CORS.

(API Overview). After reading this on the API Overview page of JASPAR, the coders began to believe that this error was due to miscoding of the **getJasparInfo()** function, but after studying the code and testing the API calls through the JASPAR interface, they realized that this issue was not due to the structure of their code, but in fact was due to JASPAR and the fact that they don’t support CORS. At this point, the JASPAR team met with the professor and decided that the best solution would be to implement a relay controller that would handle all traffic from JASPAR to GRNsight. Once the relay controller was done being integrated with the server-side of GRNsight, the API calls began working properly, which corresponded with the end of the intermediate phase of this project (Figure C).

The next phase of the project was the finishing touches (red) phase, which was where the coders spent most of their time interacting with the other teams working on this feature, to make sure that all of the pieces were working together properly (Figure C). The first step was to make sure that the JASPAR information, that the biologists on the integration team were requiring for the final page, was properly being organized and returned so that the information was useful on the page. This required taking the object returned from the second API call and properly “filing” the relevant fields into their spots within the master object. At this point, all that was left to do was to pass the information onto the page design and integration team so that they knew what to expect when calling the JASPAR function, so that they could properly put it on the page, which then marked the end of the coders work on this GRNsight feature, as displayed in Figure C.

Results

Quality Assurance/Project Manager:

Database					
	Ensembl	**JASPAR**	**NCBI**	**SGD**	**UniProt**
Field to be retrieved	Gene ID	Matrix ID	Gene ID	Gene ID (Standard, Systematic, SGD)	Gene ID
	Description/Function	Class	Locus Tag	Regulation (Regulators, Targets)	Protein Sequence
	DNA Sequence	Family	Also Known As	Interaction (Total, Physical, Genetic)	Similar Protein
	Gene Location	Sequence Logo	Chromosome Sequence (ID)	Gene Ontology Summary	Protein Type/Name
	Gene Map	Frequency Matrix	Genomic Sequence (ID)	Molecular Function	Species
			Protein Sequence (ID)	Biological Process	
				Cellular Component	

Table 1. An outline of which fields the quality assurance guild determined needed to be retrieved from each of the five databases.

As is apparent from Table 1 above, the database from which the most information fields were selected was SGD. This is due to the fact that the data that is currently being analyzed using GRNsight is primarily for the yeast genome, and SGD is a database dedicated solely to the study of yeast. In this way, this database has the most in-depth information of genes in yeast and thus provided the highest number of relevant fields of information on its database. Beyond the information that could only be obtained through SGD, the rest of the information was split up between the four other databases. JASPAR (which is the database from which this team pulled their information) is a database dedicated solely to transcription factors. Therefore, the information retrieved from this page has to do with characteristics that are important for transcription factors (such as the frequency matrix and sequence logo which indicate the starting sequence of these factors). Due to the fact that UniProt is the most well-regarded database for information on proteins, fields from this database focused on protein data (such as the sequence and related proteins). Lastly, Ensembl and NCBI are more general databases that provide information for a variety of genes for a number of different species. Some of the most basic (but vital) fields were pulled from these databases (such as gene sequence, description etc.) as they could continue to be used in the future as GRNsight's functionality expands beyond primarily just looking at data for transcription factors in yeast to looking at different types of genomic networks in all kinds of species.

In addition to this primary task, the quality assurance team member worked closely with the coders from both API teams to ensure that all of the proper fields were being pulled from the databases. While all of the fields were able to be correctly pulled from JASPAR, there were some fields that the hAPI team coders were unable to retrieve from the other four databases. A list was started regarding these fields that could not be obtained. However, this list (which as of now only includes "Similar Proteins" from UniProt) is incomplete, as the coders were still working to retrieve fields at the time this paper was written (and also because the quality assurance team member was unable to retrieve all of these fields despite reaching out to the coders of the hAPI team on multiple occasions).

Data Analyst:

ANOVA	WT	dZAP1
p < 0.05	2528 (40.85%)	2485 (40.2%)
p < 0.01	1652 (26.70%)	1609 (26.0%)
p < 0.001	919 (14.85%)	885 (14.3%)
p < 0.0001	496 (8.01%)	457 (7.4%)
Benjamini & Hochberg-corrected p < 0.05	1822 (29.44%)	1766 (28.5%)
Bonferroni-corrected p < 0.05	248 (4.01%)	209 (3.4%)

Table 2. Calculated p-values from the ANOVA test of dZAP1 data compared to the wild type (WT) p-values (Aporras1_Week_8) (Trynaur, E., 2017) .

The ANOVA test was used to determine if genes had expression change that was different than zero at any time point. As the p-value decreases, less gene expression change is considered significant and the percent (%) of genes included is decreased as displayed by the change between genes in $p < 0.05$ and $p < 0.01$ in Table 2. As the p value cutoffs are decreased, confidence in the data is increased in classifying the genes as changing expression other than zero at any time point. The Benjamini & Hochberg p-value calculation was used as a statistical tool to decrease the possibility of false discovery (Thissen, D., Steinberg, L., & Kuang, D., 2002). The Bonferroni p-value calculation was used as a more stringent criteria for the gene expression change to be significant compared to the Benjamini & Hochberg p-value which explains the percent (%) of significant genes in dZAP1 as lower, 3.4%, compared to 28.5% as displayed in Table 2.

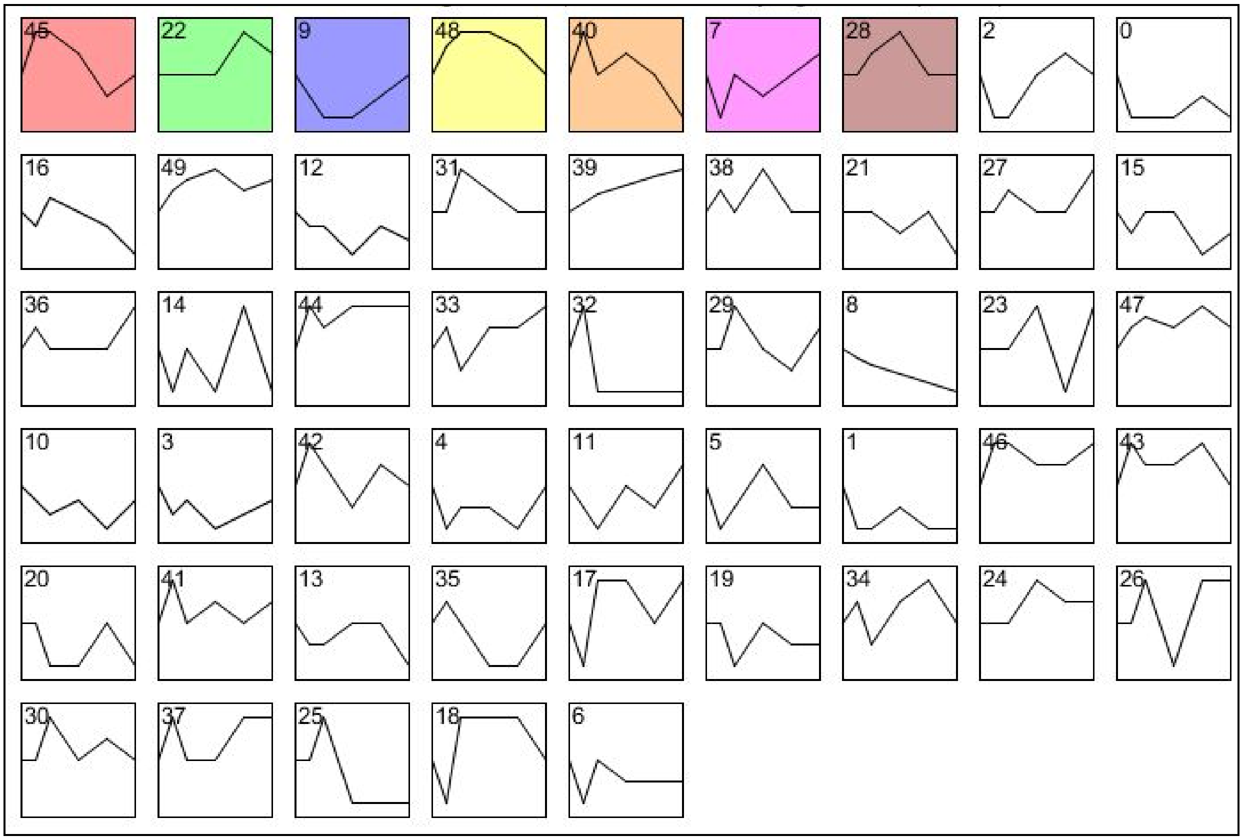


Figure E. Seven significant profiles were generated from high-level analysis STEM clustering and visualization from dZAP1 ANOVA data (Aporras1_Week_10).

STEM provided a visualization of the expression of genes within significant clusters and also generated GOlists to examine which cell process genes were changing in expression during the experiment.

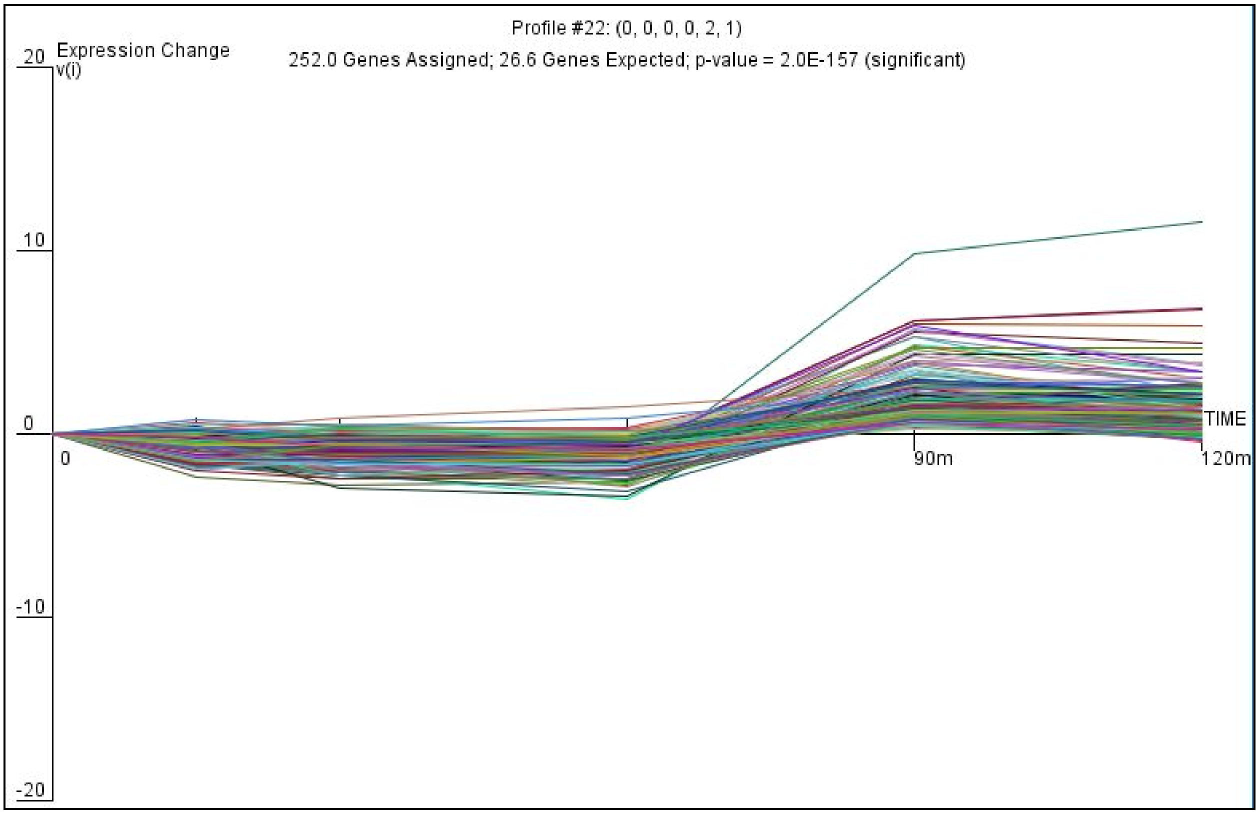


Figure F. A more in-depth visual of the expression of 252 genes within Profile 22 of dZAP1 (Aporras1_Week_10).

Profile 22 was selected because it displayed no significant change in expression until the beginning of the recovery phase after time point 60m until time point 90m where expression began to decrease. This suggests these genes were not changed in regulation, either downregulated or upregulated, during the process of sustaining cold shock. However, in recovery to the stimulus, there was an upregulation of the cluster of genes before returning back down in the direction the original expression change of zero which suggests a response in these genes in order to recover. This could a mechanism by which the cell recovers and returns to normal conditions following cold shock. STEM clustering, in Figure F, also provides the very low p-value for the profile which was calculated to be 2.0 E-157 which characterizes its significance among the other profiles. Thus, one may have confidence in the validity of the cluster expression change in profile 22 of dZAP1.

Category ID	Category Name	#Genes Category	#Genes Assigned	#Genes Expected	#Genes Enriched	p-value	Corrected p-value	Fold
GO:0003779	actin binding	19	11	2.7	8.3	9.60E-06	0.006	4.1
GO:0098805	whole membrane	127	35	17.9	17.1	2.80E-05	0.012	2.9
GO:0005739	mitochondrion	331	70	46.7	23.3	6.50E-05	0.038	1.5
GO:0098754	detoxification	26	12	3.7	8.3	7.50E-05	0.04	3.3
GO:0008092	cytoskeletal protein binding	30	13	4.2	8.8	8.40E-05	0.048	3.1
GO:0051156	glucose 6-phosphate metabolic process	11	8	1.6	6.4	1.60E-05	0.012	5.2

Table 3. Six selected GO terms from the GOlist of Profile 22 of dZAP1 generated by STEM analysis (Aporras1_Week_10).

When examining the categories more in depth though the GOlist, terms reveal cell processes and structures which were altered in the recovery phase of the experiment. Actin binding, found in Table 3 is characterized as cell processes which bind to actin or other types of filaments which are essential in cells to maintain their shape (The Gene Ontology, 2017).

Additionally in Table 3, cytoskeletal protein binding presents a similar mechanism which is associated with proteins of the cytoskeletal network which also includes actin filaments as discussed before (The Gene Ontology, 2017). The whole membrane consists of the phospholipid bilayer and any proteins, integral or peripheral, and any chance in gene expression in the recovery phase would signify that the cell's membrane became more active (The Gene Ontology, 2017). Both mitochondria and glucose 6-phosphate are extremely important to the cell in terms of generating ATP. Specifically, the mitochondria is where respiration occurs in which ATP is synthesized and glycolysis is the first phase of cellular respiration (The Gene Ontology, 2017). Glucose 6-phosphate is the second molecule in glycolysis produced from hexokinase and glucose and is a step in the process of generating pyruvate for the cell to direct to the citric acid cycle. Compared to the other categories, glucose 6-phosphate metabolic processes displayed the largest fold of 5.2 as is presented in Table 3. Therefore, the cell is upregulating genes which are responsible for cellular processes which generate ATP, pyruvate, and ultimately energy for the cell after the cold shock. Finally, detoxification was another category defined and described as a any processes of the cell which are oriented towards removing toxic substances (The Gene Ontology, 2017). During the cold shock stimulus, the cell may have been susceptible to releasing toxic substances usually stored safely within organelles to destroy polymers and dispose of them properly as is characteristic of lysosomes.

Transcription Factor	ABF1	ACE2	ASH1	CIN5	CST6	GCN4	GLN3	HAP4
p-value	1.58E-7	1.69E-8	1.13E-9	4.62E-5	2.14E-5	5.48E-8	.513787	2.12E-8
Transcription Factor	MSN2	MSN4	RPN4	SFP1	SOK2	YAP1	ZAP1	
p-value	1.48E-11	4.73E-12	2.95E-12	9.66E-10	1.91E-9	1.39E-9	2E-15	

Table 4. Transcription factors and their p-values generated from YEASTRACT (Aporras1_Week_14).

With the use of the Profile 22 genelist, YEASTRACT was used to generate a list of transcription factors which target the genes within the cluster from Profile 22. Although 49 were considered significant, only 15 were selected for GRNsight visualization as displayed in Table 4 with their respective p-values. These were selected because of their significant, low, p-values and with the assistance from the professor who determined that the number of transcription factors to be used for visualization was appropriate (Aporras1_Week_15).

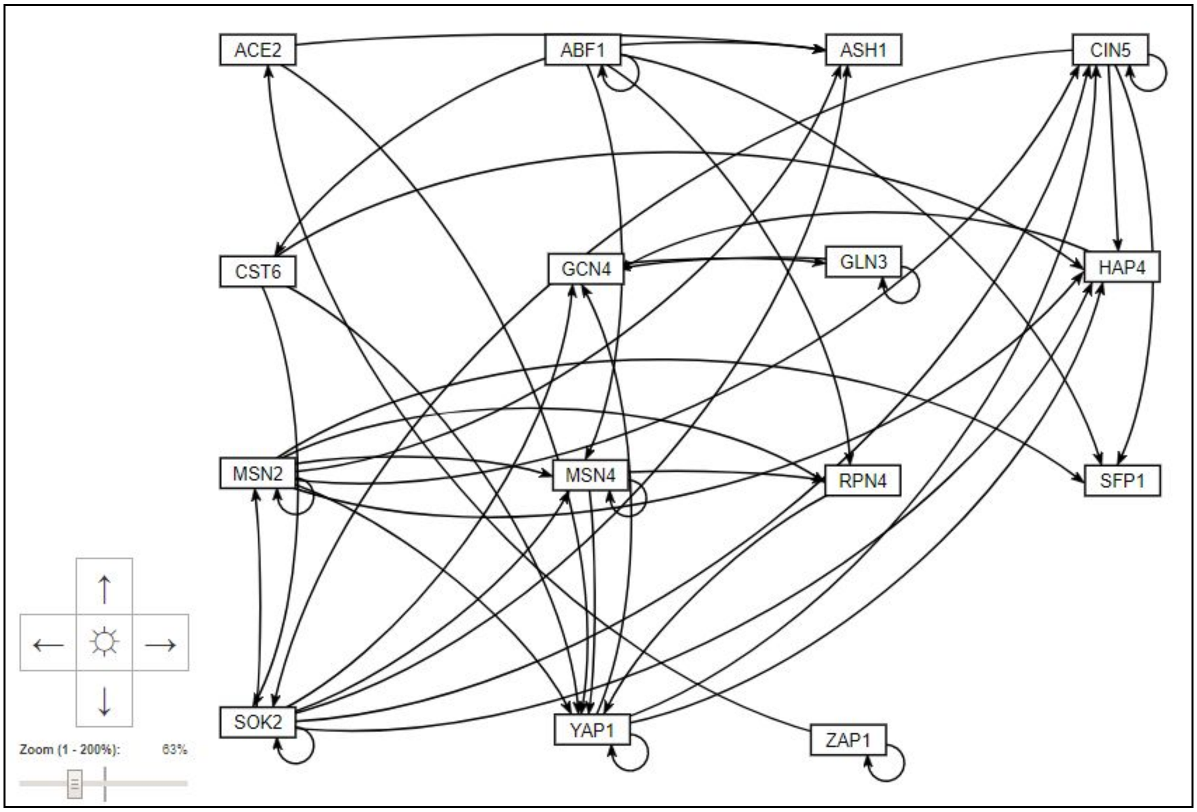


Figure G. Unweighted transcription factor regulatory network generated from GRNsight (Aporras1_Week_15).

Once the adjacency matrix and network were created from YEASTRACT's regulation matrix, the network was then input into GRNsight for visualization (Aporras1_Week_15). When examining Figure G, MSN2 appears to be regulating many of the other transcription factors within the network which could be signifying its importance in the cluster in terms of stress responses. Additionally in Figure G, YAP1, HAP4, and CIN5 all have at least four (4) transcription factors regulating their individual expression. Therefore, this could infer their regulation is integral in mediating gene expression between the target genes and the transcription factors which regulate these specific transcription factors of YAP1, HAP4, and CIN5.

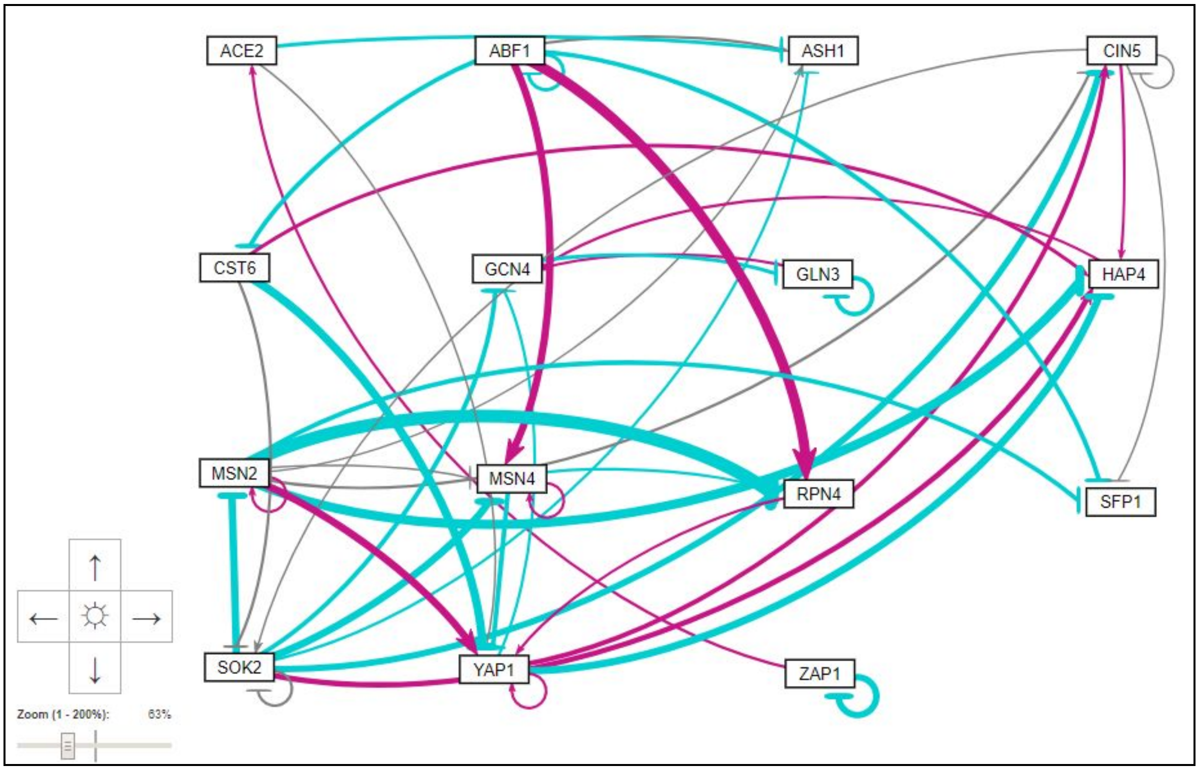
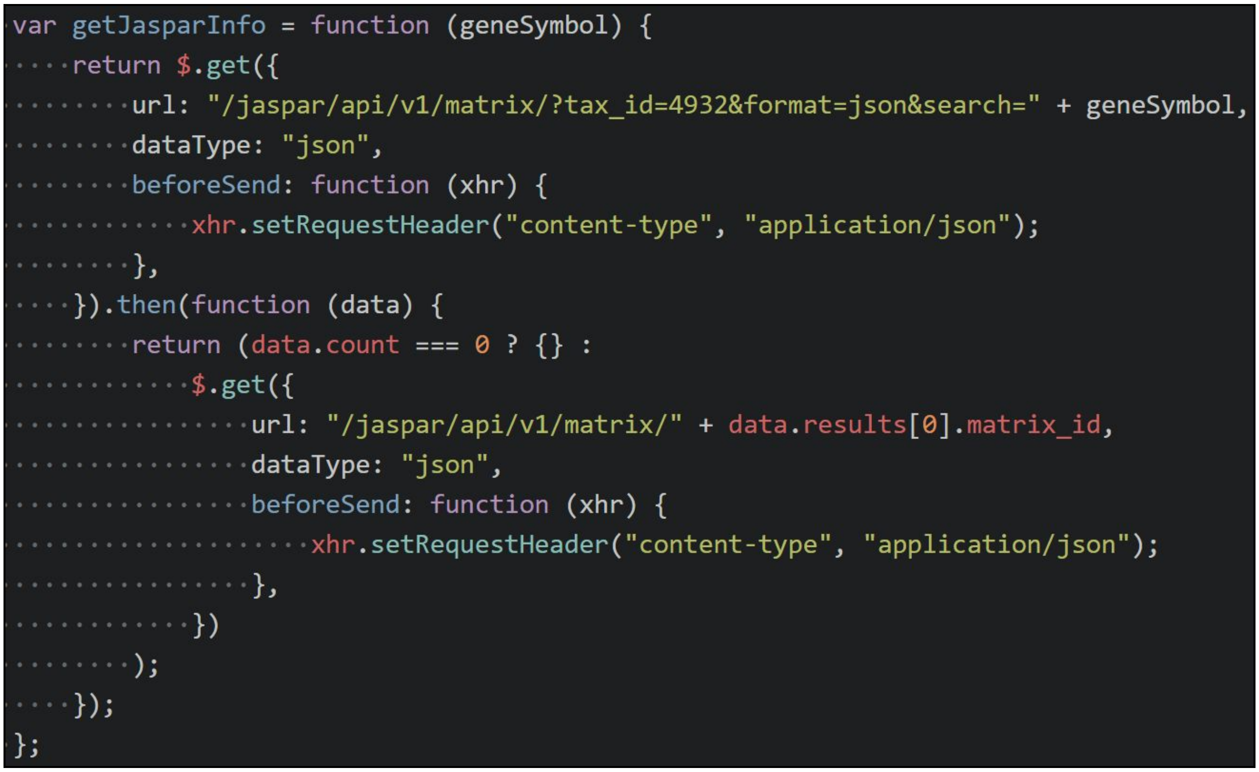


Figure H. Weighted transcription regulatory network generated from GRNsight where pink signifies upregulation and blue signifies downregulation (Aporras1_Week_15).

MATLAB was used to generate an output which would allow GRNsight to display the direction of regulation, upregulation or downregulation, and the magnitude of regulation of transcription factors (Aporras1_Week_15). Displayed in Figure H, most of the regulation between the transcription factors are characterized as upregulation (pink) compared to downregulation (blue). RPN4 has a very large magnitude upregulation and large downregulation which could infer it requires significant input required to turn it "on" and similarly significant input to turn it "off". With most of the regulation being upregulation in Figure H, this appears to be congruent with the significant increased fold of genes in the recovery phase following cold shock as it attempts to recover from the stimulus.

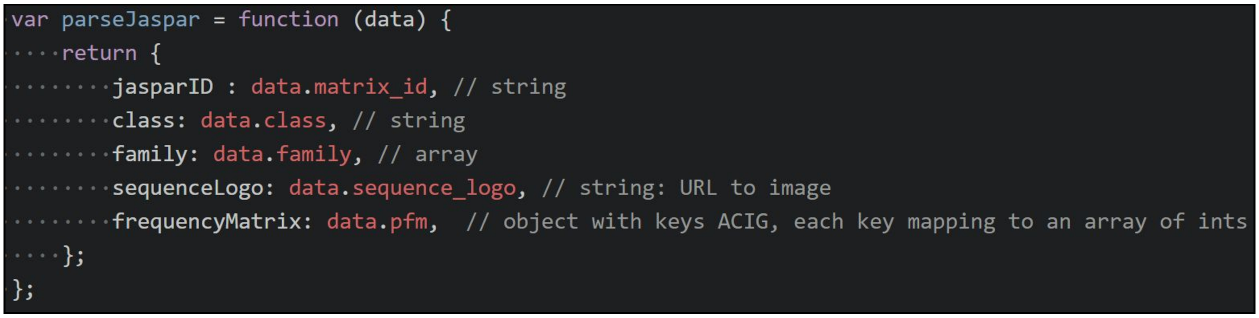
Coders:



```
var getJasparInfo = function (geneSymbol) {
    return $.get({
        url: "/jaspar/api/v1/matrix/?tax_id=4932&format=json&search=" + geneSymbol,
        dataType: "json",
        beforeSend: function (xhr) {
            xhr.setRequestHeader("content-type", "application/json");
        },
    }).then(function (data) {
        return (data.count === 0 ? {} :
            $.get({
                url: "/jaspar/api/v1/matrix/" + data.results[0].matrix_id,
                dataType: "json",
                beforeSend: function (xhr) {
                    xhr.setRequestHeader("content-type", "application/json");
                },
            })
        );
    });
};
```

Figure I: API calls work together to retrieve gene information object from JASPAR database.

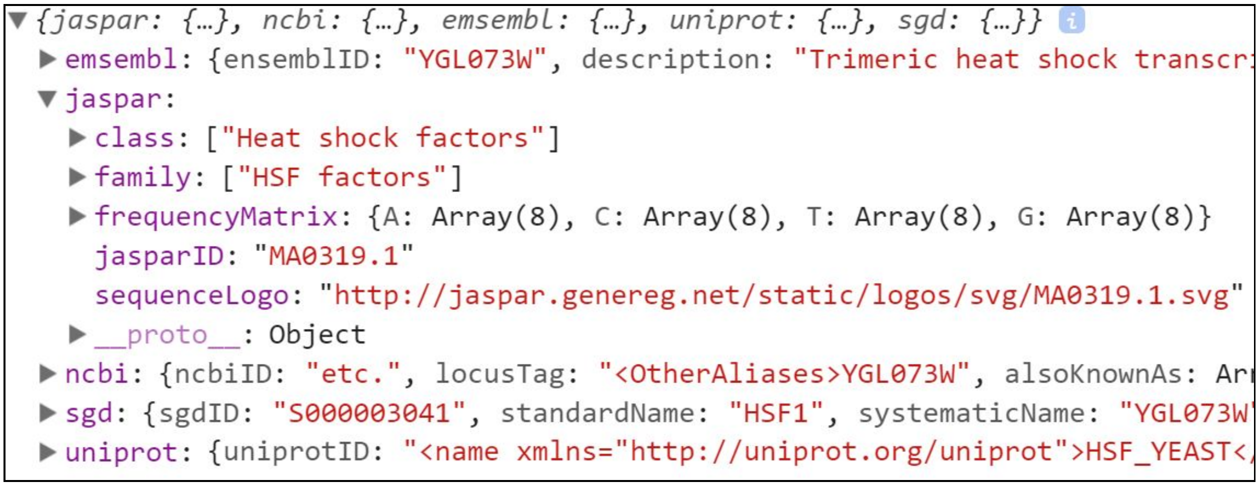
In the figure above, you will find the bulk of the work that the coders of the JASPAR team have spent their time on for this project (Figure I). The getJasparInfo() function consists of two parts: retrieve a list of search results and retrieve an object using a matrix ID. It has already briefly been explained up above, but these two API calls work in unison to produce the resulting object. The first call produces an array of search result objects, each object coming with a slight description of what you will find on that gene's page. Within this array, one can take the first index, which is the value that most accurately represents the gene symbol being asked for, and retrieve it's matrix ID. This matrix ID is JASPAR's way of distinguishing between the different genes and species that they have stored within their database. JASPAR also has an API call which requires a matrix ID to return an object will all of the information that they have about that gene. The coders called that API call and passed in the retrieved matrix ID, and this resulting in the object that was needed.



```
var parseJaspar = function (data) {
    return {
        jasparID : data.matrix_id, // string
        class: data.class, // string
        family: data.family, // array
        sequenceLogo: data.sequence_logo, // string: URL to image
        frequencyMatrix: data.pfm,  // object with keys ACIG, each key mapping to an array of ints
    };
};
```

Figure J: JASPAR information that is returned from Figure I is parsed into masted object.

Next, the coders took the information that was returned from the function in Figure I, and filtered out all of the extraneous data that was unnecessary to this feature, either because it was already being retrieved from a different database or because the biologists of our team didn't find it necessary to be displayed on the gene page. The function in Figure J does most of this work by only adding to itself everything that was asked upon, and ignoring the rest.



```
▼ {jaspar: {…}, ncbi: {…}, emsembl: {…}, uniprot: {…}, sgd: {…}} i
  ▶ emsembl: {ensemblID: "YGL073W", description: "Trimeric heat shock transcr
  ▼ jaspar:
    ▶ class: ["Heat shock factors"]
    ▶ family: ["HSF factors"]
    ▶ frequencyMatrix: {A: Array(8), C: Array(8), T: Array(8), G: Array(8)}
      jasparID: "MA0319.1"
      sequenceLogo: "http://jaspar.genereg.net/static/logos/svg/MA0319.1.svg"
    ▶ __proto__: Object
  ▶ ncbi: {ncbiID: "etc.", locusTag: "<OtherAliases>YGL073W", alsoKnownAs: Ar
  ▶ sgd: {sgdID: "S000003041", standardName: "HSF1", systematicName: "YGL073W
  ▶ uniprot: {uniprotID: "<name xmlns="http://uniprot.org/uniprot">HSF_YEAST</
```

Figure K: Final product, master object that is returned when one calls getGeneInformation()

This last figure shows how all of the different databases are compiled into one master object, which has everything very clearly labeled. Within the master object, one will find a key for each of the databases that were used. And the value of each of these is itself an object, containing all of the information that was retrieved from that page. This is the master object that was then passed onto the other teams.

Conclusion

Through this experiment, further insight was gained regarding the effect of cold shock on yeast. While past research has found there to be significant change in the expression of a number of genes in yeast when exposed to cold shock (Becerra et al., 2003), there has been little knowledge accumulated regarding which transcription factors belong in the regulatory network for the cold shock response of yeast. This experiment, which was composed of two distinct portions, seeked to both utilize and enhance the regulatory network mapping technology of GRNsight (Dahlquist et al., 2016), to both better understand the relationship between transcription factors in yeast's cold shock response and enhance the functionality of the GRNsight tool for future research.

In regards to the first portion of this study, GRNsight was used to map the regulatory network of gene ontology profile 22 of the dZAP1 strain of yeast. This profile was chosen as it was one whose expression was significantly impacted by yeast's cold shock response. A first primary result can be found by looking at the results of the GO terms of this profile in the context of past research which found that the cold shock regulatory response in yeast was similar to the heat shock response, and thus proposed that this response was a general stress response (Becerra et al., 2003). With this research in mind, it can then be inferred that functions such as actin binding and detoxification, in Table 3, are general stress responses for the yeast cell that are not specific to cold shock. Furthermore, looking at the graph of expression for genes in profile 22 (Figure F), one can see that the actual change in expression for these genes did not occur until time point 60, at which point the cold shock had been halted and the recovery period had begun. This further supports the idea that the regulatory responses in yeast cells are not specific to cold shock, but rather are general stress responses that work to return the cell to homeostasis after the environmental stress has subsided. Finally, from the analysis of the gene regulatory network (Figure G & Figure H), one can see that the majority of the interactions between the transcription factors was that of repression, suggesting that the majority of these general stress responses that are part of the cell's response to cold shock are a result of a downregulation of a number of transcription factors.

Turning now to the second part of the experiment, which focused on enhancing GRNsight through the addition of an informative gene page for each gene node, the JASPAR the Friendly Ghost team worked to incorporate information from the JASPAR database into this page. The results from JASPAR the Friendly Ghost's team project may be described as an object that contains all of the relevant data requested by the project managers/biology majors (Simonwro120_Week_14). This data was acquired through two API queries, which were sent to the JASPAR API in a function named "getJasparInfo" (Simonwro120_Week_15).

Paul Ford writes in part six of his article "What is Code?" about how applications are made, which can be found in the *Bloomberg Businessweek* magazine. This section of the article, titled "How Are Apps Made?", highlights a myriad of different ideas, tools, and environments which facilitate and expedite the process of creating an application (Ford, 2015). Of all these ideas, tools, and environments mentioned, APIs (Application Programming Interfaces), text editors, and the idea of version control are the three most related concepts discussed in the paper in reference to JASPAR's project results (Simonwro120_Week_11). Application Programming Interfaces were very relevant to the JASPAR's coding team and their results (JASPAR_the_Friendly_Ghost). In fact, the objective of JASPAR's coding team was to return the specified data of a certain gene. This was accomplished by writing queries to the JASPAR API with the correct specifications in the “getJasparInfo” function. The environment that the JASPAR’s coding team worked in was a text editor called “Atom”. Atom is a simple text editor application which allows one to open up and access the code of their files. This is where the bulk of the coding took place for the JASPAR coding team (Simonwro120_Week_14). This is an integral part of creating any application and arguably the hardest and most time consuming in many cases. Almost all projects that involve working with a large and complicated application like GRNsight involve multiple people working on different aspects of the project. When one codes in an environment that involves many people all working on the same application, they will need a way to control and record the changes that were made to the code. In the unfortunate event that the application stops running, this recorded list of changes can help the programmers identify and fix the bug (Ford, 2015). This practice is referred to as version control, which may be implemented through the use of GitHub, and is essential for large groups of coders writing code for the same application. In this way, a number of industry techniques were utilized in the integration of JASPAR information into the new gene page feature of GRNsight. Overall, this retrieved JASPAR data will provide transcription factor specific information to the new gene page feature (JASPAR, 2017), which as a whole will allow GRNsight users the ability to obtain basic information regarding the genes in a regulatory network without having to leave the GRNsight application.

In conclusion, this research project worked to utilize the features that are already in place on GRNsight to gain a better understanding of gene regulatory networks specific to yeast’s cold shock response, while also working to build a new GRNsight informational gene page feature that will enhance the application for future users. Going forward, there are a number of further steps that could be taken to expand upon this research. In regards to studying yeast genomic response to cold shock, additional profiles could be analyzed to see if they follow similar trends to the profile 22 analyzed in this study. Furthermore, data from different studies on the topic of yeast's cold shock response could be analyzed alongside results from Dr. Dahlquist's lab to see if they return similar results and GRNsight regulatory network maps. In terms of the added GRNsight feature, due to the time constraints of this project, further testing and the addition of more information to the gene pages are areas of potential future work. Isolating the code from the server using *Sinon* would ensure more reliable functionality (Sinon, 2017), while modifying the code to check for any servers that are not responding (and returning a message to the user if this is the case) would create for a better user interface. Finally, there were a few fields that were unable to be retrieved due to various reasons, and while a complete list of these fields has not yet been able to be obtained (as the coders were still working on the code at the time of this paper be written), once this list is accumulated, finding ways to retrieve this information through workarounds or other databases is another area for potential improvement of the new GRNsight feature.

Acknowledgments

The JASPAR the Friendly Ghost team would like to thank both of the professors, Dr. Dahlquist and Dr. Dionisio, for the work they've put into instructing the class and supporting the students both through the final project and throughout the entire semester. Additionally, thank you to the other teams and peers who the JASPAR the Friendly Ghost team had the incredible opportunity of working alongside with on this project; with the ultimate goal of adding a new feature to GRNsight and gaining a better understanding of regulatory networks of *saccharomyces cerevisiae*. Thank you to LMU Biology Department and Computer Department for providing the resources necessary throughout the course instruction and the space to drive forward knowledge concerning biological databases and bioinformatics. Finally, to the JASPAR the Friendly Ghost team for a cohesive effort through the last weeks of the semester to complete the goals of the project.

References

About GRNsight. (2017, June 2). *GRNsight*. Retrieved on December 7, 2017, from http://dondi.github.io/GRNsight/about.html

Actin binding. The Gene Ontology. Retrieved November 29, 2017, from http://amigo.geneontology.org/amigo/term/GO:0003779

Becerra, M., Lombardia, L. J., Gonzalez-Siso, M. I., Rodriguez-Belmonte, E., Hauser, N. C., & Cerdan, M. E. (2003). Genome-wide analysis of the yeast transcriptome upon heat and cold shock. *Comparative and Functional Genomics, 4*(4), 366-375. doi: 10.1002/cfg.301

Cytoskeletal protein binding. The Gene Ontology. Retrieved November 29, 2017, http://amigo.geneontology.org/amigo/term/GO:0008092

Dahlquist, K. D., Dionisio, J. D., Fitzpatrick, B. G., Anguiano, N. A., Varshneya, A., Southwhick, B.J., & Samdarshi, M. (2016). GRNsight: a web application and service for visualizing models of small- to medium-scale gene regulatory networks. *PeerJ Computer Science* 2:e85 https://doi.org/10.7717/peerj-cs.85

DeRisi, J. L., Iyer, V. R., & Brown, P. O. (1997). Exploring the Metabolic nd Genetic Control of Gene Expression on a Genomic Scale. *Science, 278*(5338), 680-686. doi: 10.1126/science.278.5338.680

Detoxification. The Gene Ontology. Retrieved November 29, 2017, from http://amigo.geneontology.org/amigo/term/GO:0098754

Ford, P. (2015). What Is Code? Part 6: How Are Apps Made? Retrieved on December 10, 2017, from https://www.bloomberg.com/graphics/2015-paul-ford-what-is-code/#how-are-apps-made

Gene Ontology Consortium. (2017). The Gene Ontology. Retrieved December 13, 2017, from http://geneontology.org

Glucose 6-phosphate metabolic process. The Gene Ontology. Retrieved November 29, 2017, from http://amigo.geneontology.org/amigo/term/GO:0000054

JASPAR Rest API. (2017). API Overview. Retrieved December 14. 2017, from http://jaspar.genereg.net/api/overview/

JASPAR Rest API. (2017). JASPAR API. Retrieved December 14. 2017, from http://jaspar.genereg.net/api/

JASPAR. (2017). About JASPAR. Retrieved on December 14, 2017, from http://jaspar.genereg.net/about/

LMU BioDB 2017. (2017). Aporras1 Week 10. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Aporras1_Week_10

LMU BioDB 2017. (2017). Aporras1 Week 14. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Aporras1_Week_14

LMU BioDB 2017. (2017). Aporras1 Week 15. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Aporras1_Week_15

LMU BioDB 2017. (2017). Aporras1 Week 8. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Aporras1_Week_8

LMU BioDB 2017. (2017). Coder. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Coder

LMU BioDB 2017. (2017). EBachour Week 11. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/EBachoura_Week_11

LMU BioDB 2017. (2017). EBachour Week 12. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/EBachoura_Week_12

LMU BioDB 2017. (2017). EBachour Week 14. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/EBachoura_Week_14

LMU BioDB 2017. (2017). EBachour Week 15. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/EBachoura_Week_15

LMU BioDB 2017. (2017). JASPAR the Friendly Ghost. Retrieved December 05, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/JASPAR_the_Friendly_Ghost

LMU BioDB 2017. (2017). QLanners Week 11. Retrieved December 9, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/QLanners_Week_11

LMU BioDB 2017. (2017). QLanners Week 12. Retrieved December 9, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/QLanners_Week_12

LMU BioDB 2017. (2017). QLanners Week 14. Retrieved December 9, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/QLanners_Week_14

LMU BioDB 2017. (2017). QLanners Week 15. Retrieved December 9, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/QLanners_Week_15

LMU BioDB 2017. (2017). Simonwro120 Week 11. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Simonwro120_Week_11

LMU BioDB 2017. (2017). Simonwro120 Week 14. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Simonwro120_Week_14

LMU BioDB 2017. (2017). Simonwro120 Week 15. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Simonwro120_Week_15

LMU BioDB 2017. (2017). Week 10. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Week_10

LMU BioDB 2017. (2017). Week 8. Retrieved December 13, 2017, from https://xmlpipedb.cs.lmu.edu/biodb/fall2017/index.php/Week_8

Mitochondrion. The Gene Ontology. Retrieved November 29, 2017, from http://amigo.geneontology.org/amigo/term/GO:0005739

Short Time-series Expression Miner (STEM). (2006). Retrieved December 13, 2017, from http://www.cs.cmu.edu/~jernst/stem/

Thissen, D., Steinberg, L., & Kuang, D. (2002). Quick and easy implementation of the Benjamini-Hochberg procedure for controlling the false positive rate in multiple comparisons. *Journal of Educational and Behavioral Statistics*, *27*(1), 77-83.

Tyranuer, E. (2017, October 23). *WT_p-value_slide_ET* [PPTX]. Los Angeles: Emma Trynauer.

University of Michigan. (2009). Yeast biology yields insight into human knowledge expansion. *Phys.org*. Retrieved on December 14, 2017, from https://phys.org/news/2009-03-yeast-biology-yields-insights-human.html

Using yeast in biology. (2016). Retrieved December 13, 2017, from https://www.yourgenome.org/stories/using-yeast-in-biology

Whole membrane. The Gene Ontology. Retrieved November 29, 2017, from http://amigo.geneontology.org/amigo/term/GO:0098805

Yourgenome. (2016). Using yeast in biology. Retrieved on December 11, 2017, from https://www.yourgenome.org/stories/using-yeast-in-biology