Exploring the Yeast Genome with Generalized Singular Value Decomposition

Andrew Ferguson

Advisor: Professor Alexandre d'Aspremont

Submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science in Engineering

Department of Operations Research and Financial Engineering

Princeton University

June 2008

I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

_____

Andrew Ferguson

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Andrew Ferguson

Abstract

The method of generalized singular value decomposition (GSVD) is used to identify the principal components separating and shared between DNA microarray time courses of the yeast *Saccharomyces cerevisiae* under two different experimental conditions. In the first analysis, a comparison is performed between the yeast stress response to hydrogen peroxide (H_2O_2) and the stress response to the drug menadione (MD). The analysis confirms the similarity between the responses on a genome-wide scale. Furthermore, GSVD is shown to successfully cluster the genes involved in the oxidative stress response (OSR), the expression profile of which is confirmed to be affected by the choice of stress agent. In the second analysis, the gene expression profiles of yeast undergoing H_2O_2 stress are compared with that of yeast growing under normal conditions. The decomposition again identifies the genes involved in the OSR.

Acknowledgements

I would like to thank Professor Olga Troyanskaya for letting me take her graduate class in the spring of 2007 in which I was introduced to the GSVD by presenting the Alter paper. I would also like to thank Matt Hibbs for providing several yeast time series data sets which proved to be a pleasure to work with computationally, and Max Staller for the primer on yeast and the stress response.

To Fannie, my family, and my friends

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The application of generalized singular value decomposition (GSVD) in a biological setting was pioneered in Alter et al. (2003a). A form of principal components analysis (PCA), GSVD is a linear algebraic technique developed by the signals analysis community for comparing two high-dimensional, time-varying signals. It is used to separate the components unique to and shared between the two signals (Golub and Loan (1996)). This project uses GSVD to explore hypotheses about the cellular stress-response of the model organism *Saccharomyces cerevisiae*.

1.1 Mathematical background

When analyzing a signal, it is often useful to break down the original signal into distinct components. These components can then be individually explored, or their combined structure examined. One particularly useful inquiry is to determine which components are strongest; these components are known as the principal components. If the original signal is a noisy analog signal, removing all but the principal components helps to eliminate noise. If the original signal is very high dimensional, determining the principal components of such a signal reveals the basic structure and main global features of the input. For this reason, PCA is considered a dimensionality

reduction technique. In the case of microarray data, it is the behavior of the global features which we want to understand.

Many methods for feature analysis exist in the signals community. Examples include Fourier Analysis, Wavelet Analysis, and matrix decomposition. In matrix decomposition, the discrete signal is first written as a vector at each time point, forming a matrix when concatenated together. Next, the matrix is decomposed so that it is transformed into a new, more informative basis. A more informative basis is one in which the principal components of the signal are revealed.

In the case of a square matrix, PCA corresponds to the eigenvalue decomposition. The eigenvectors are the components of the signal, and their eigenvalues indicate how much of the original signal each accounts for. The larger the eigenvalue, the more important the corresponding feature. The eigenvectors with the largest eigenvalues are considered the principal components.

For a non-square matrix, the concept of the eigenvalue decomposition is extended to the singular value decomposition (SVD). In the thin SVD defined by Golub and Loan (1996), the $m \times n$ matrix $\mathbf{M}$ is written $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ is an $m \times n$ unitary matrix, $\mathbf{V}$ is an $n \times n$ unitary matrix, and $\mathbf{\Sigma}$ is an $n \times n$ diagonal matrix. The values along the diagonal of $\mathbf{\Sigma}$ are known as the singular values, and serve the same role as the eigenvalues in PCA. The columns of $\mathbf{V}$ form a basis of input features, while the rows of $\mathbf{U}$ indicate how much of each input feature makes up each dimension of the original signal.

Finally, if we decompose two different input signals, but require that they share the same basis of input features, we develop the GSVD. Since the two signals share the same input basis, we can readily compare the relative strengths of the shared principal components.

1.1.1 Generalized Singular Value Decomposition

GSVD is a linear algebraic method of diagonalizing two rectangular matrices ($N_1 \times M$ and $N_2 \times M$) into M principal components and computing the significance of each component in the first matrix relative to the second. We will use the following definition of the GSVD: If the first gene expression matrix is $\mathbf{E}_1$ ($N_1 \times M$) and the second matrix is $\mathbf{E}_2$ ($N_2 \times M$), then the GSVD transform satisfies:

$$\begin{aligned} \mathbf{E}_1 &= \mathbf{U}_1\mathbf{S}_1\mathbf{X} \\ \mathbf{E}_2 &= \mathbf{U}_2\mathbf{S}_2\mathbf{X} \end{aligned}$$

where $\mathbf{X}$ is the M genelets $\times$ M arraylets basis of inputs, $\mathbf{S}_i$ ($M \times M$) is the positive semi-definite diagonalized form of $\mathbf{E}_i$, and $\mathbf{U}_i$ ($N_i \times M$) is the representation of each row of $\mathbf{E}_i$ in the new basis. $\mathbf{U}_1$ and $\mathbf{U}_2$ are both orthogonal matrices. Note that the dimensions of the matrices produced by the decomposition depend upon the GSVD definition chosen. (Golub and Loan (1996))

In this project, we will require that genelets, the vectors which make up the rows of $\mathbf{X}$, be normalized. Formally, we write this as the condition:

$$\|\langle m|\mathbf{X}\| = 1 \qquad \forall \quad 1 \leq m \leq M$$

where $\langle m|\mathbf{X}$ is the m^{th} row vector of $\mathbf{X}$.

The generalized singular values are the ratios of the elements of $\mathbf{S}_1$ and $\mathbf{S}_2$, which are both diagonal matrices. In general, when $\mathbf{E}_2$ is an invertible square matrix, the generalized singular values are equal to the regular singular values of the matrix $\mathbf{E}_1\mathbf{E}_2^{-1}$. The generalized singular values are unique up to a rearrangement of the rows of $\mathbf{X}$ (the genelets). We can think of them as the global "gain controls" on the genelets. The columns of $\mathbf{X}$ will be called the arraylets.

1.1.2 Analyzing the results of a GSVD

After performing a principal components analysis, it is important to study two relationships: the relationship between the components as a whole to the original data signal, and the relationship between the principal components. For example, if a data signal is actually composed of ten independent, identically-distributed components, a PCA of only five of the components will not capture the majority of the variance in the original signal. Furthermore, if we analyzed the ten principal components of that same data signal, we would see that each component would account for roughly equal proportions of the overall variance.

The most informative measure of the degree to which the data sets have been separated by the GSVD calculation is the antisymmetric angular distance between the components of the data sets. This angular distance is defined as

$$\theta_m = \arctan(\mathbf{S}_{1,m}/\mathbf{S}_{2,m}) - \pi/4.$$

The value of θ_m indicates the significance of the mth genelet in the first data set relative to the second and ranges from $\pm\pi/4$ (indicating significance in one data set only) to 0 (indicating equal significance). (Alter et al. (2003a))

Next, we turn our attention to the relationship of the components to a single signal. To measure the degree to which each genelet (m) explains the fluctuations in the two data sets separately, Alter et al. (2003b) introduced the "generalized fractions of eigenexpression" defined as

$$P_{i,m} = \mathbf{S}_{i,m}^2 / \sum_{k=1}^{M} \mathbf{S}_{i,k}^2, \quad \text{where } i = 1,2.$$

That is, $P_{1,2}$ is the fraction of variance in the first signal explained by the second

principal component. It also used the "generalized normalized Shannon entropy,"

$$D_i = \frac{-1}{\log(M)} \sum_{k=1}^{M} P_{i,k} \log(P_{i,k}), \quad \text{where } i = 1,2.$$

which measures the amount of disorder in each data set. A value of $D_i = 0$ indicates that all variance is explained by one component, while a value of $D_i = 1$ indicates that all components contribute equally to the overall variance.

1.2 Biological background

Since the discovery of the structure of deoxyribonucleic acid (DNA) in 1953, a driving force in molecular biology has been the effort to decode the meaning of the genetic code. Today's model of cellular biology describes a process in which DNA is *transcribed* onto messenger ribonucleic acid (mRNA). The mRNA is then *translated* by ribosomes into amino acids, which are the building blocks of proteins, the biological structures which serve as the cellular machinery. The section of DNA that codes for the making of a particular protein is known as a gene. The level (or amount) of mRNA copies present in the cell for a particular gene is called the level of gene expression. The decision to transcribe certain genes and not others is known as the process of gene regulation. (Brent (2000))

Being biological compounds, proteins break-down over time. Thus, cells are constantly replenishing the supply of heavily-used proteins. Additionally, cells respond to many internal and external events by increasing or decreasing the quantity of various proteins through gene regulation. Therefore, by observing which genes are being transcribed at a given time, biologists can infer the level of each protein in the cell. We should note upfront however, that this inference is not perfect. Proteins which respond to an external event could have been stockpiled in the past, or the transcription of a given gene could merely be a preventative measure – the resulting protein is not

guaranteed to be involved in the cellular response (Schulze and Downward (2000)). However, due to the aforementioned break-down, it is widely accepted that the level of gene expression strongly correlates with the levels of protein activity within the cell. Techniques for measuring the levels of protein presence directly are currently in development (e.g., Kislinger et al. (2006)), but are still primitive and expensive.

1.2.1 Measuring gene expression

Measuring the levels of gene expression in a cell (the "expression profile") is done using a device known as a DNA microarray, or "genechip." A microarray consists of a glass, silicon or nylon substrate with thousands of wells, each one containing a specific single-stranded section of DNA. To measure the expression profile, transcription is blocked (arrested) in the target cell (or colony of cells) and the target DNA is freed from the cell and poured over the genechip. The DNA then hybridizes with the chip's DNA sections – that is, the freely-moving target DNA bonds with its fixed complementary strand. Thus, through this process, the targetted genes have been spatially separated. By fluorescently tagging the target DNA prior to this separation, biologists can then measure the quantity of DNA in each well by measuring the fluorescent intensity.

The process of determining the fluorescent intensity can be done in two different manners. In the first manner, the genetic profile of two different cells (e.g., sample and control) are compared on one chip. The DNA from the sample is tagged with red dye, and the DNA from the control is tagged with green. The observed wavelength of the spot on the chip is a function of the ratio of the expression level for that gene. In the second process, a single class of cells is used (e.g., just the sample) and the absolute level of fluorescence is measured. The two methods are known as "two-color" and "one-color," respectively. The DNA microarray was first presented in Schena et al. (1995). An example of a two-color microarray is presented in Figure 1.1.

The analysis of data from DNA microarrays has exploded during the last thirteen

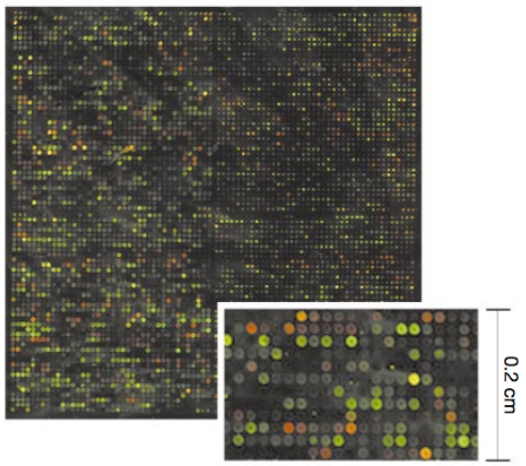


Figure 1.1: An example two-color microarray from Lockhart and Winzeler (2000).

years – Google Scholar reports more than 5,000 citations of the original paper at the time of this writing. Numerous statistical problems exist in this space, such as how to normalize the fluorescence data (the red and green dyes are not equally strong emitters), how to compare the results of two-color studies with one-color studies, how to eliminate background noise and experimental bias (such as patterns due to the method of pouring the DNA over the gene chip), and how to account for purely biological noise such as DNA mis-match or incompletely separated genes. An additional hurdle to exploiting the DNA microarray technique is the high dimensionality of the data. A simple yeast study with two-conditions and 10 microarrays under each condition generates more than 90,000 data points.

DNA microarray data is thus a prime candidate for exploring general dimensionality-reduction techniques. Many papers use clustering techniques such as k-Means or hierarchical clustering to organize the data for further analysis. Another approach is to use signal analysis PCA techniques such as singular value decomposition (SVD) or GSVD.[1] After all, the gene expression profile is a time-dependent function of the

[1]We should note that k-Means has been shown to be a variant of PCA (Ding and He (2004)). However, the use of non-clustering approaches to microarray analysis is still relatively novel.

gene and cell conditions. The use of SVD has begun to catch-on in the microarray community (e.g., Alter et al. (2000), Wall et al. (2001), Yeung et al. (2002)). However, sometimes we wish to compare the expression levels of two organisms or one organism under two conditions – generalized singular value decomposition, described below in Section 1.1.1 gives us a framework for doing precisely that.

1.2.2 The yeast stress response

The yeast *S. cerevisiae* has been studied under a variety of stress conditions, such as heat shock, nutrient starvation, hyper-osmotic shock, hypo-osmotic shock, and oxidative stress (Gasch et al. (2000), Saldanha et al. (2004)). The study by Gasch et al. (2000) found that the yeast cell, when faced with any of more than ten different stresses, responds globally in the same way as measured by the gene expression profile. This shared response to all stresses was termed the environmental stress response (ESR). The Gasch experiments found that about 600 genes (mostly involved in cell and DNA replication) were all repressed in the ESR, while about 300 genes involved in "carbohydrate metabolism, detoxication of reactive oxygen species, cellular redox reactions, cell wall modication, protein folding and degradation, DNA damage repair, fatty acid metabolism, metabolite transport, vacuolar and mitochondrial functions, autophagy, and intracellular signaling" were all induced in the ESR. In other words, yeast cells consistently respond to stress by slowing growth (stopping the cell-cycle – the process by which cells reproduce themselves) and fortifying against the stress. The cell-cycle consists of four stages: G1, in which the cell grows normally; S, in which the cell's DNA is replicated for the daughter cell; G2, in which the cell again grows normally and prepares for division; and finally the M stage (mitosis), in which the cell splits in two.

Oxidative stress can be induced through different means, including exposure to hydrogen-peroxide (H_2O_2) and the drug menadione (MD). Shapira et al. (2004) com-

pared the mRNA expression profiles over time of separate yeast cultures experiencing oxidative stress in these two manners. That study confirmed the presence of the ESR as described by Gasch et al. (2000), and isolated the type and (limited) portion of each response that was different under the two triggers of oxidative stress. With GSVD's ability to compute the *relative* significance of components, could it be used to reach the same conclusions?

Before we apply the algorithm, we must recognize that the gene profile signals from H_2O_2 and MD are very similar because they are both yeast responses to oxidative stress. Thus, we should not expect GSVD to produce a very strong separation between the two signals – a strong separation would be inconsistent with what we know about both the inputs and previous experimental results. We hope, however, that it will highlight the limited difference enough to provide guidance for future experiments.

A second reasonable application of GSVD would be to compare the gene profile of yeast undergoing H_2O_2-stress with that of yeast undergoing normal cell growth. The standard time series of gene expression during the cell cycle is presented in Spellman et al. (1998). We hope that GSVD will be able to separate the genes whose regulation is a reaction to the oxidative stress. In the Alter et al. (2003a) paper which introduced GSVD to the microarray community, the technique was applied to the yeast cell-cycle profiles from Spellman et al. (1998) and the human cell-cycle profiles from Whitfield et al. (2002). In that study, GSVD was extremely successful at identifying similar and unique genetic themes present in the expression profiles. A 6-dimensional basis was chosen from the 18 components present in the decomposition; this new basis well-approximated the behavior of the 4,523 genes in the yeast data and the 12,056 genes in the human data.

1.2.3 Gene ontology

The strength of PCA techniques is their ability to rapidly reduce the dimensionality of very large data sets. Although you can run PCA techniques on any data set of your choice, the results are not necessarily physically or biologically meaningful. The components which are identified may simply be the result of mathematical organization. Therefore, in order to report biologically relevant findings, it is necessary to perform further tests, either with other computational techniques or with classical lab work.

One computational technique that can be used to explore biological significance is to check the principal components for statistically-signficant enrichment of gene ontological (GO) terms. GO terms are a standardized set of labels established by the Gene Ontology Consortium. There are three major classifications of GO terms: cellular component (e.g., ribosome, nucleus), biological process and molecular function. Curated databases associating the genes of a single organism are maintained online. As scientists positively identify genes with particular GO terms, the entry for that gene is updated with the new GO terms. The database of yeast GO terms is located at http://www.yeastgenome.org. (Ashburner et al. (2000))

Testing for GO term enrichment is a straight-forward statistical test. If a particular gene clustering is not biologically significant, then the distribution of GO terms from genes in that cluster should match the overall (or background) distribution. However, if more genes in the cluster share a GO term than would be expected *a priori*, the cluster is said to be biologically significant and enriched for that specific GO term.

Chapter 2

Experimental Methods

The raw data from Shapira et al. (2004) and Spellman et al. (1998) were obtained from the SPELL (Serial Pattern of Expression Levels Locator) project presented in Hibbs et al. (2007). The SPELL project provides a search tool for comparisons across numerous yeast microarray data sets. As such, all data available from the project has been normalized for comparison across different experimental protocols (such as the use of one-color Affymetrix chips versus two-color Eppendorf chips). The data sets were also preprocessed by SPELL to impute missing values using the k-Nearest Neighbor algorithm. Additionally, technical replicates (that is, the same DNA hybridized to more than one array) were averaged together. The use of the cleaned and well-maintained data from the SPELL project insulated this project from the statistical issues described previously in Section 1.2.1.

For the comparison between the H_2O_2 and MD trials, the tab-delimited data contained in the file `Shapira04.flt.knn.avg.pcl` was loaded into `MATLAB` using the `Import Data` command. For the comparison between the H_2O_2 stress-response and the unstressed yeast cell-cycle, the files `Shapira04.flt.knn.avg.pcl` and `Spellman98_alphaFactor.flt.knn.avg.pcl` were processed with the Python program `common.py` presented in Appendix A to identify the 4,287 genes common to

both datasets. The resulting files, `Shapira04.flt.knn.avg.pcl.common` and `Spellman98_alphaFactor.flt.knn.avg.pcl.common` were also loaded into `MATLAB` using the `Import Data` command for tab-delimited data. See the code comments in Appendix A for more information about the imported data.

For both experiments, the equations presented in Alter et al. (2003a) and Alter et al. (2003b) were implemented in the source files listed in Appendix A – the `MATHEMATICA` code presented in Alter et al. (2003b) is not generalized for use in other studies.

Analysis of the actual GSVD calculation, presented in Section 3.1, closely follows the model presented by Alter et al. (2003a). The analysis in Sections 3.2-4.2 was specific to this project. Gene Ontology (GO) terms were analyzed using the *Saccharomyces* Genome Database project's GOTermFinder, which is available online at http://db.yeastgenome.org/cgi-bin/GO/goTermFinder.

The choice of the Shapira et al. (2004) data set, and the specific arrays analyzed, was guided by the warning in Alter et al. (2003a) that the "one-to-one correspondence between the two sets of conditions is at the foundation of the GSVD comparative analysis of the two data sets and should be mapped out carefully." There are four time courses available in the Shapira et al. (2004) data set, two of which, H_2O_2 II and MD II, were performed under identical conditions, except for the choice of oxidative stress-exerting agent. Both samples were placed in G1 arrest at time $t = 0$ and had the stress-exerting agent introduced at $t = 35$ minutes. Microarray analysis was performed on the H_2O_2 sample at $t \in \{0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 100, 120, 140, 160, 180\}$ minutes and on the MD sample at $t \in \{0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 77, 85, 100, 120, 130, 140\}$. Thus, GSVD was performed on the time points in the intersection of these two sets, for a total of 14 time points, $t \in \{0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 77, 100, 120, 140\}$ minutes.

A second GSVD analysis was performed to compare the Shapira et al. (2004)

H_2O_2 II data set with the Spellman et al. (1998) data set of the yeast cell-cycle (synchronized by α-factor arrest). In the figures and code presented in this paper, the Spellman et al. (1998) data is referred to as "Base" since it represents a baseline at which the cell is not experiencing any environmental stresses. Microarray analysis was reported for this sample at $t \in \{0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 84, 98, 105, 112, 119\}$ minutes. The GSVD analysis was performed on the 13 common time points, $t \in \{0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 98, 119\}$ minutes.

The Spellman et al. (1998) α-factor synchronized data set is the same yeast cell-cycle data set used in the Alter et al. (2003a) paper.

Chapter 3

Analysis

3.1 Results of the first decomposition

Graphical depictions of the decomposition of the two Shapira et al. (2004) time course arrays are presented in Figures 3.1 and 3.2 for the H_2O_2 and MD conditions, respectively. This is a decomposition from the original 4,524 genes $\times$ 14 arrays space to the diagonalized 14 genelets $\times$ 14 arraylets space.

After performing the decomposition, we first examine the statistics described in section 1.1.2 for information about the identified components. The values of the anti-symmetric angular distance for each component, denoted θ_m for $m \epsilon \{1, 2, ..., 14\}$, are shown in Figure 3.3.

Alter et al. (2003a) used the cutoff $\|\theta_m\| \geq \pi/8$ to indicate which genelets are highly significant. Under this criteria, no genelets from this decomposition can be regarded as highly significant. As mentioned previously, this is an outcome consistent with what we might expect *a priori* given that the two data sets are quite similar, having undergone similar forms of stress. Furthermore, this is consistent with the results published in Shapira et al. (2004), which state that the differences in the expression profiles under these two conditions can be accounted for by "two small

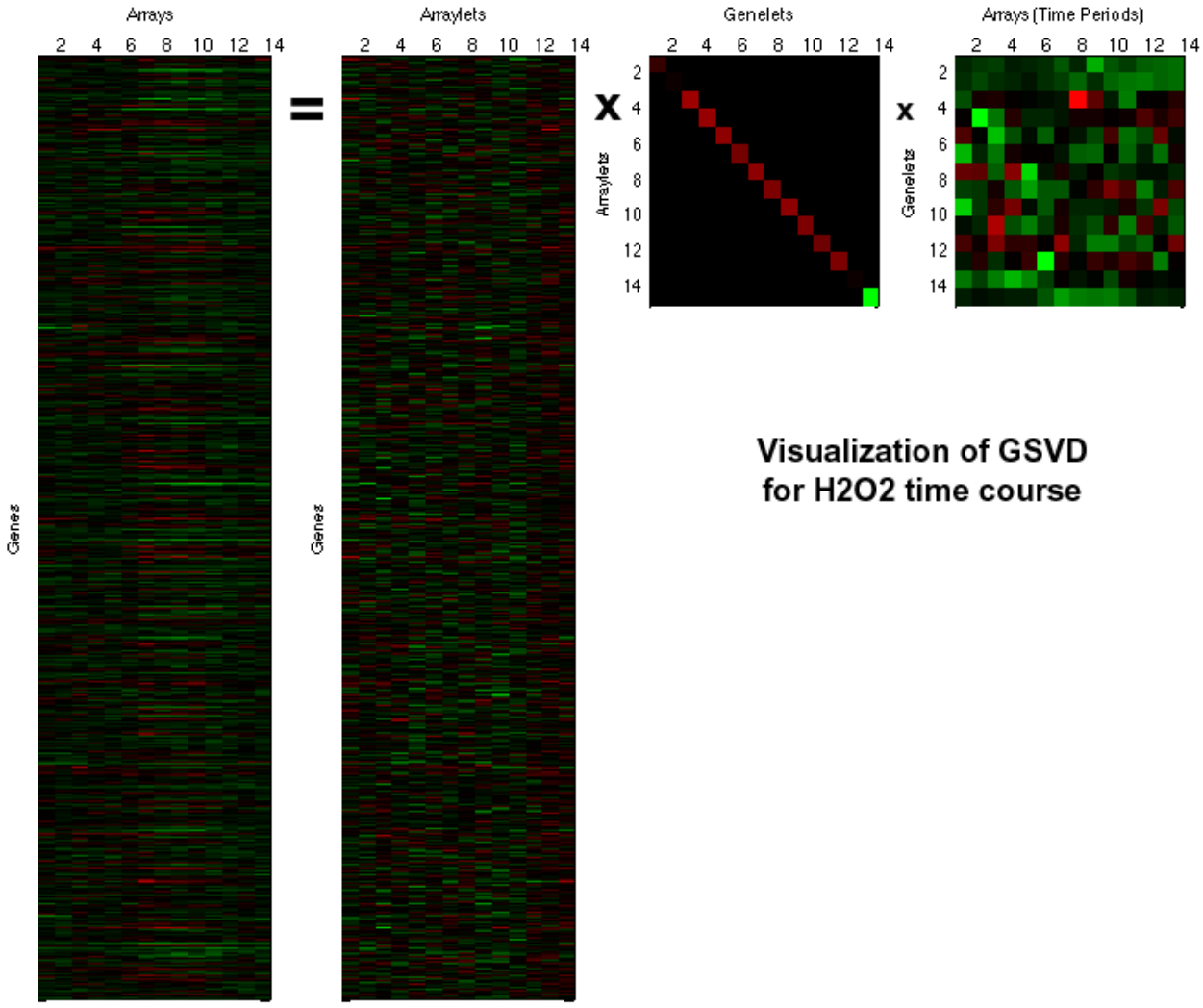


Figure 3.1: Generalized singular value decomposition (GSVD) of the H_2O_2 data set.

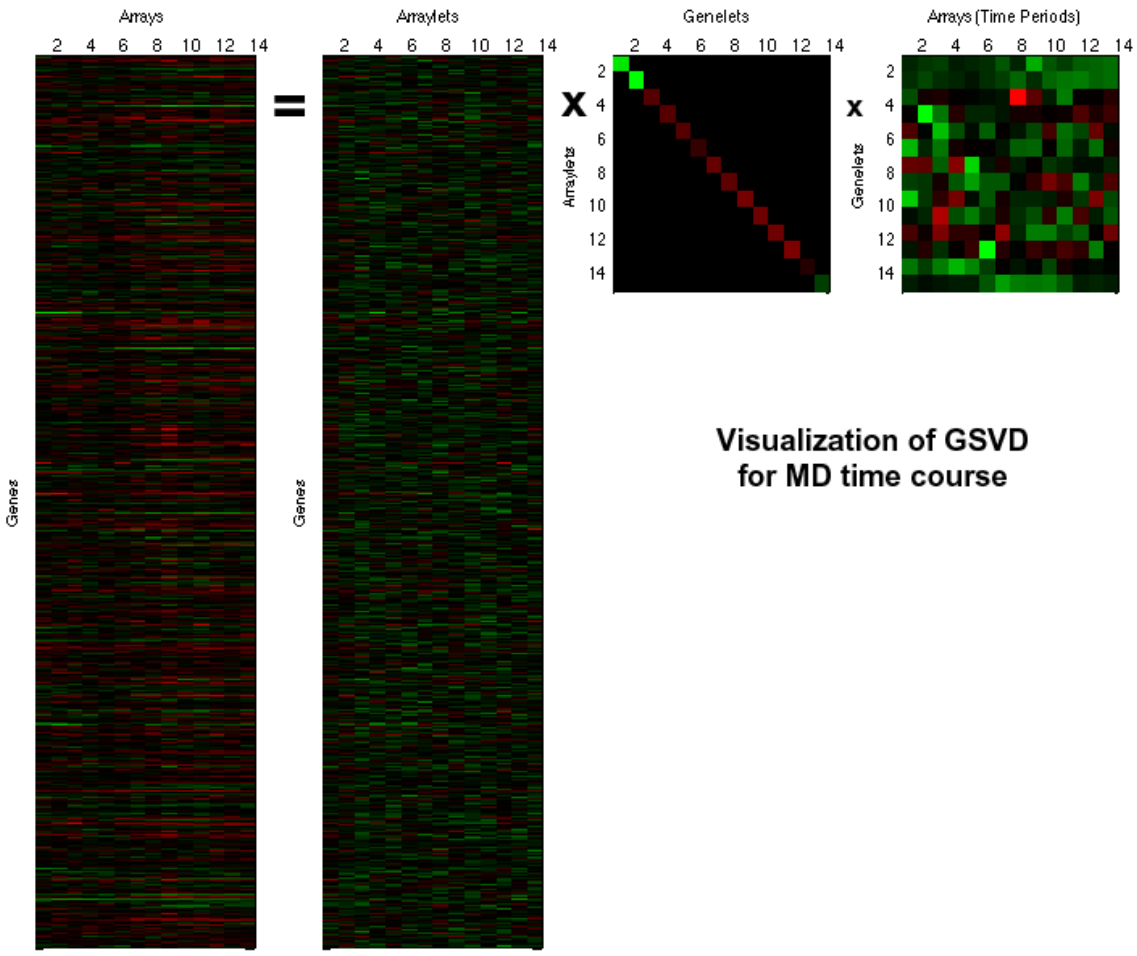


Figure 3.2: Generalized singular value decomposition (GSVD) of the MD data set.

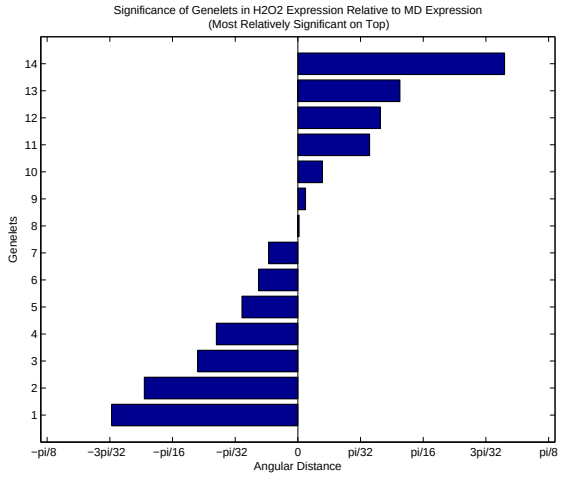


Figure 3.3: Angular distances for the H_2O_2, MD decomposition.

coexpressed groups of genes" regulated by a single complex. Computing these angular distances will guide our search when we look for groups of genes that are regulated differently under the two conditions.

One of the main applications of principal component analysis (PCA) such as SVD is to group the data by how much of an affect it has on the overall system. These learned groupings are the principal components. In a noisy signal with a few driving forces, a successful application of PCA would reveal the small number of components that account for the majority of the variance, and a larger number of components that continue to explain the data set's variance, each to a lesser extent.

As described previously, we can calculate the relative strength of each of the fourteen shared components in the two different signals. The generalized fractions of eigenexpression and normalized Shannon entropy are presented in Figure 3.4 for the H_2O_2 data set and Figure 3.5 for the MD data set. As expected, neither of these measures indicate that the GSVD computation strongly separated the two data sets. The result that both cases have fairly large normalized Shannon entropies ($D_1 = 0.74878$, $D_2 = 0.80439$) confirms that the yeast's response is very, very similar to both H_2O_2-induced and MD-induced oxidative stress.

3.2 Exploring the genelets

GSVD can be viewed as a form of clustering. Each gene can be considered as belonging to the specific genelet which accounts for the greatest proportion of variance within that gene. For GSVD, this would be the genelet that is the most parallel (or antiparallel) to the gene when the gene is projected onto the basis formed by the genelets. In the GSVD expression $\mathbf{E}_1 = \mathbf{U}_1\mathbf{S}_1\mathbf{X}$, the values in the i^{th} row of $\mathbf{U}_1$ express this projection for the i^{th} gene. In that row, the column with the largest absolute value corresponds to the genelet which explains the greatest proportion of

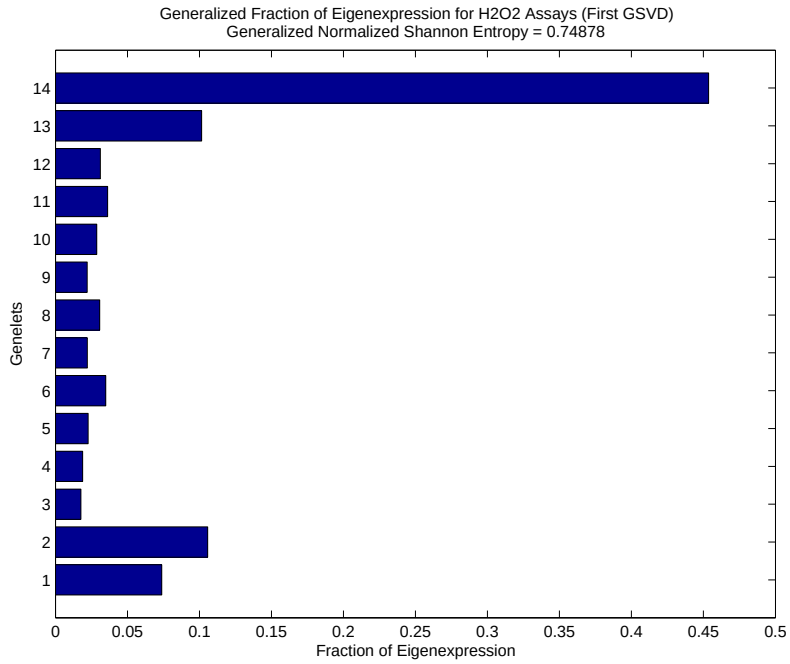


Figure 3.4: Generalized fraction of eigenexpression and normalized Shannon entropy ($D_1 = 0.74878$) for the H_2O_2 data set.

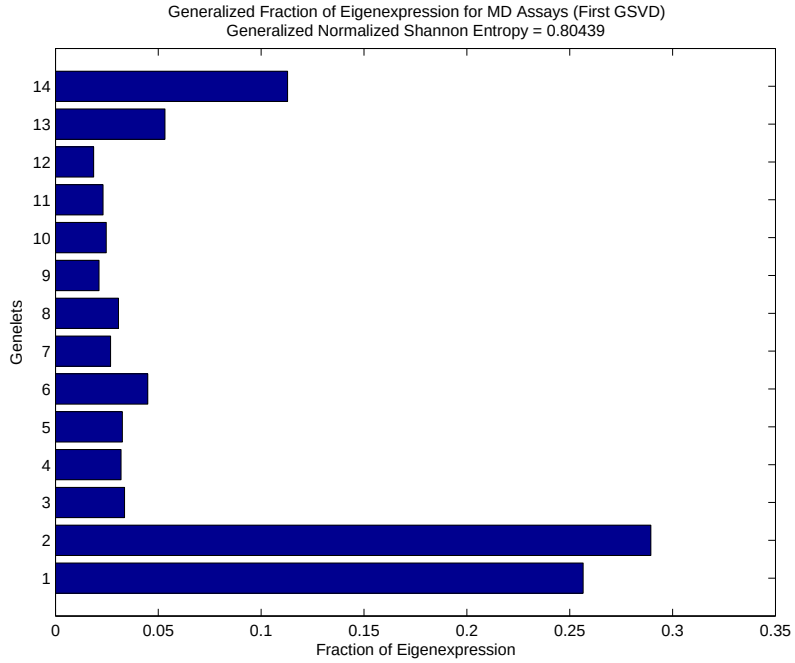


Figure 3.5: Generalized fraction of eigenexpression and normalized Shannon entropy ($D_2 = 0.80439$) for the MD data set.

variance.

Using this natural clustering that is created, clustered maps of the H_2O_2 and MD expression data are presented in Figure 3.6. Few strong clusters jump out of this clustering. This is most likely due to the large number of clusters created (28 – one parallel and one antiparallel for each of the fourteen genelets). If we compare the clustered figures below with the unclustered figures above (on the left of Figures 3.1 and 3.2), the organization is apparent.

When we turn the question around and examine the genelets individually, several interesting patterns appear. For each genelet, the genes were ranked by the magnitude of their component in the direction of that genelet, first using the expression profiles of the H_2O_2 data, and then using the expression profiles of the MD data. The expression profiles were then displayed for the top 200 genes in both the parallel and antiparallel directions. Finally, next to each profile display, the profile of those same genes under the other experimental condition is shown. Clustering in this manner highlights the genes which had the largest variance between the two data sets. Identifying those genes which were regulated differently under the two conditions is the first step towards understanding the cell's global response to different oxidative stresses.

For example, Figure 3.7 shows the expression profiles for the top four hundred genes in the H_2O_2 data most aligned with genelet 14. These are shown on the left-hand side of the figure. On the right-hand side, we see for those same genes their expression profile in the MD data set. Genelet 14 appears to contain a group of genes which are part of a potentially cyclic process whose cycle is either disrupted or less coherent when oxidative stress is induced by MD. The fact that genes in the bottom half of the figure display an expression profile that is essentially a mirror image of the genes in the top half of the figure is highly suggestive of the idea that genelet 14 has captured some cyclic process inside the cell. It is important to note this genelet had the greatest fraction of eigenexpression in the H_2O_2 data (see Figure 3.4) and had

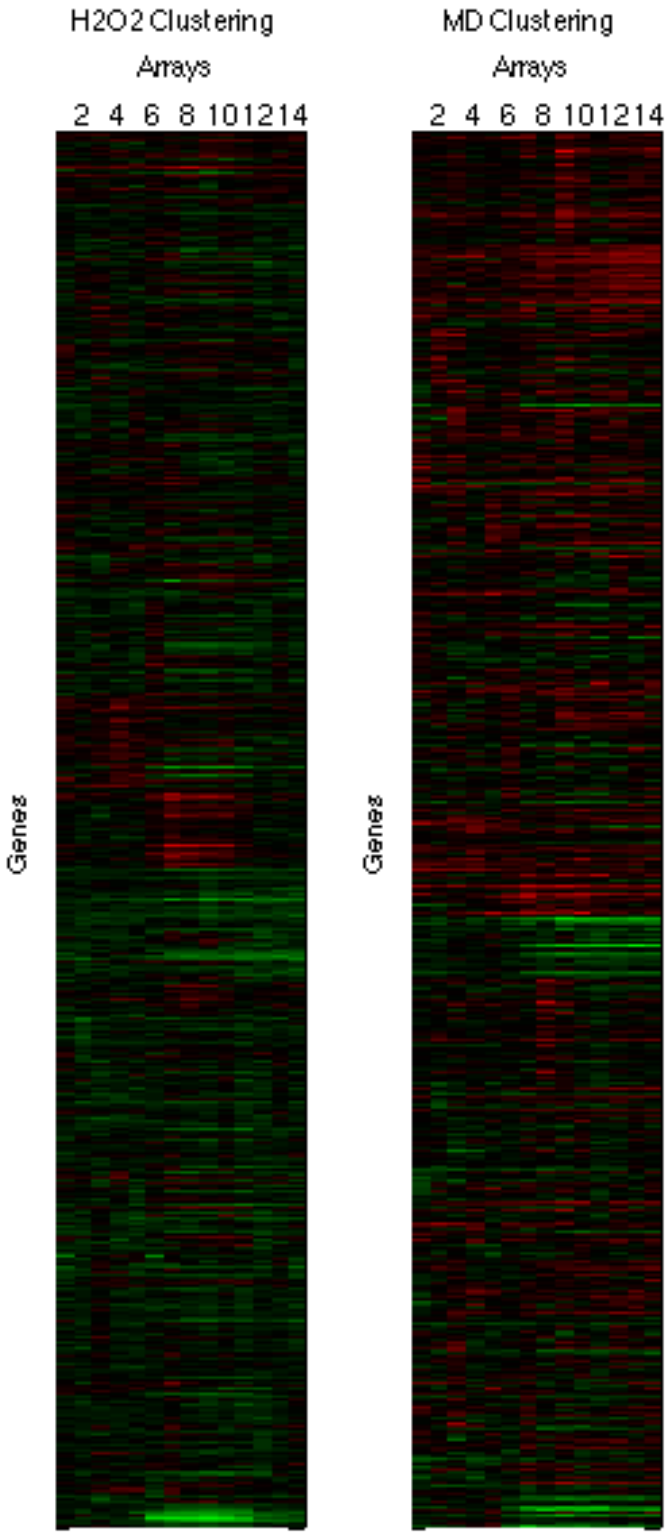


Figure 3.6: Clustering of the raw H_2O_2 and MD expression profiles using the 28 clusters created by the genelets. Each cluster consists of the genes whose greatest proportion of variance is (anti)parallel to one of the 14 genelets.

the greatest significance in the H_2O_2 data relative to the MD data (Figure 3.3).

Going further in our examination of genelet 14, we refer to Figure 3 in Shapira et al. (2004), which is reproduced below as Figure 3.8. Except for flipping the top and bottom clusters, the two figures look virtually identical. Genelet 14 is almost certainly the oxidative stress response. Figure 3.7 consists of genes which cycle once over the course of the two-hour experiment, but whose cycle becomes decoherent in the presence of menadione. Successfully capturing the oxidative stress response, a biologically-relevant distinguishing feature, is an important validation of the use of the GSVD technique in this setting. We will look deeper into the composition of genelet 14 in the next section.

By viewing the data in this way, it is easy to see which genes behave similarly in each data set and which behave differently between the two data sets. This separation is what we hoped to achieve by performing GSVD, and would allow a researcher to focus new research on the genes of interest. Lab techniques such as gene knockout, in which the genes in question are deleted and the overall effect on the organism is observed, are very effective for confirming the role of important genes. In an organism with more than 4,500 genes, gene knockout studies are only feasible when they can be targeted by computational techniques such as this.

The complete set of these figures for each genelet is given in Appendix B. By flipping through them, one can visualize the meaning of the angular distances displayed in Figure 3.3. Genelets with smaller $\|\theta_m\|$'s are in the middle of the genelet series, and even the most significant genes in these genelets tend to have a profile in the H_2O_2 data that is similar to their profile in the MD data. Genelets with larger values of $\|\theta_m\|$ contain genes with dissimilar profiles under the two conditions.

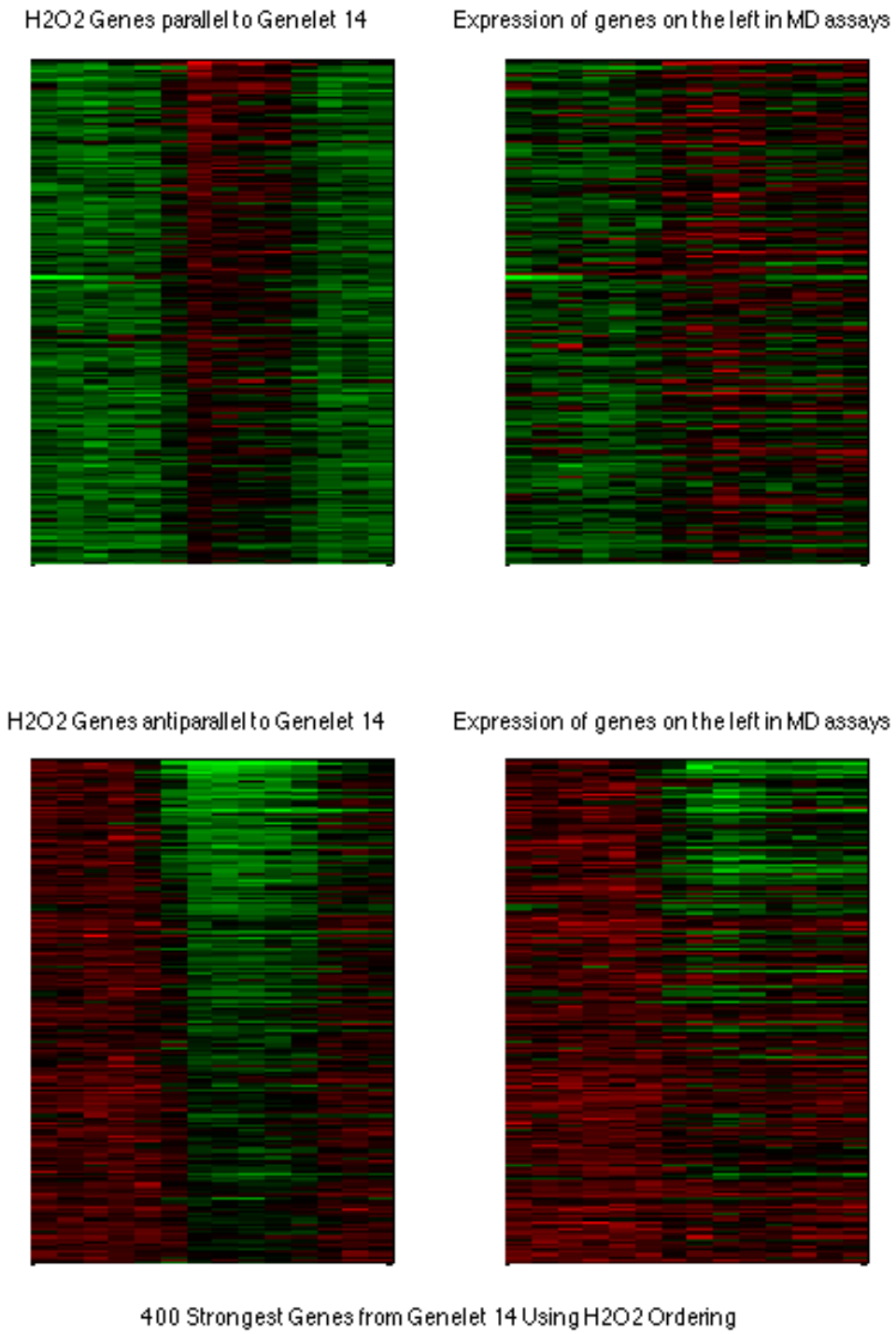


Figure 3.7: On the left, expression profiles of the 400 genes from the H_2O_2 data set with the largest (anti)parallel component along genelet 14. On the right, the expression profiles of those same genes, but in the MD dataset. The pattern of expression in this genelet is strikingly similar to the pattern of expression in the oxidative stress response shown in Figure 3.8.

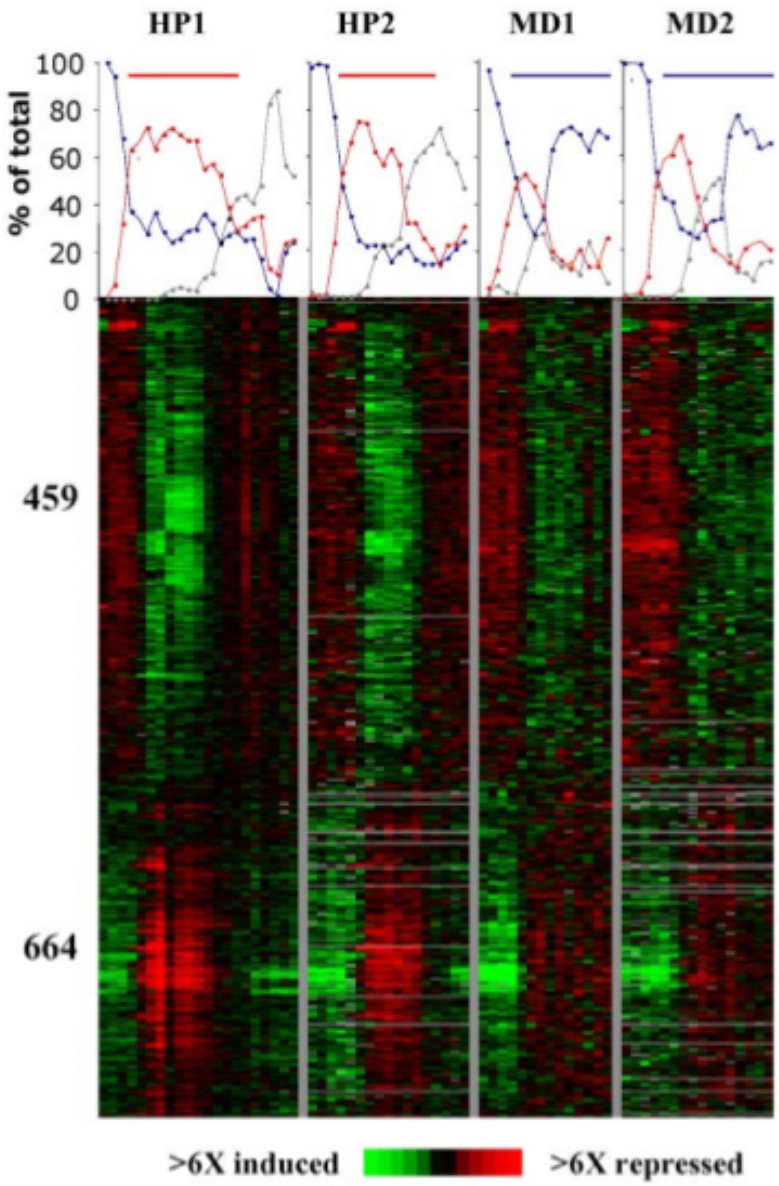


Figure 3.8: Reproduction of Figure 3 from Shapira et al. (2004). "The oxidative stress cluster. Top, cell cycle progression of treated cultures used for expression analyses. Shown are percentages of G1 (blue), S (red), and G2/M (gray) cells out of the total counted. Bottom, overview of the shared oxidative stress transcriptional response. Genes included (see Web Supplement) are those chosen based on hierarchical clustering and visual inspection of the entire ltered data set that respond both to MD and to HP. For each gene in each time course, separately, expression values were median centered to bring out the expression patterns and to assist visual comparison between different time courses in which different reference mRNA batches were used. HP1 is the same time-course experiment presented in Figure 1A. The color scale used to represent variations in transcript abundance is shown in the key at the bottomof the gure. Gray represents missing values."

3.3 Searching for GO terms

The names of the four hundred genes that had the strongest association (200 parallel, 200 antiparallel) with each genelet were saved to individual text files; one for each genelet in both the H_2O_2 data set and the MD data set. Names referring to fragments, non-open reading frames, and other non-GO terms were removed, leaving between 380 and 393 GO terms for each genelet. The files were then submitted to the GOTermFinder located at http://db.yeastgenome.org/cgi-bin/GO/goTermFinder to search for the enriched presence of GO terms in the genelets.

As described previously in section 1.2.3, the GOTermFinder reports an enrichment if the submitted listing of genes contains a statistically significantly greater amount of an ontological class of genes than would be expected based on a random sampling of genes. Here, statistical significance is defined as a p-value of < 0.01. A table of GO term enrichments that were reported follows in Table 3.1.

About 25% of each set of terms were listed as "process unknown," and even more were listed as "function unknown." Few hits came back from the GO term search. This is consistent with the GO term search performed in Shapira et al. (2004) on the original clustering of the data. The database of gene ontology terms is curated with a stringent standard of proof. Therefore, if even a few genes in our cluster (but still at a statistically significant level) are annotated to a particular GOTerm, it is reasonable to hypothesize that the other genes may be involved the same or a similar process. The successful combination of the GSVD clustering and the GOTermFinder is a strong tool for developing "guilt by association" hypotheses.

GO term searches were also run on the Component Ontology, exact results unreported. Several of the genelets exhibited enrichment for intracellular organelles, cell membrane, and other generic classes of cellular components.

3.4 Results of the second decomposition

The second application of GSVD was designed to continue probing the yeast oxidative stress response. The Shapira et al. (2004) H_2O_2 II time course was compared to the baseline unstressed yeast cell-cycle time course from Spellman et al. (1998). Once again, GSVD was able to successfully separate the oxidative stress response. Examining the generalized fractions of eigenexpression for the H_2O_2 decomposition (Figure 3.9), we see that genelet 13 explains approximately half of the variance. Could genelet 13 be the cluster we are looking for?

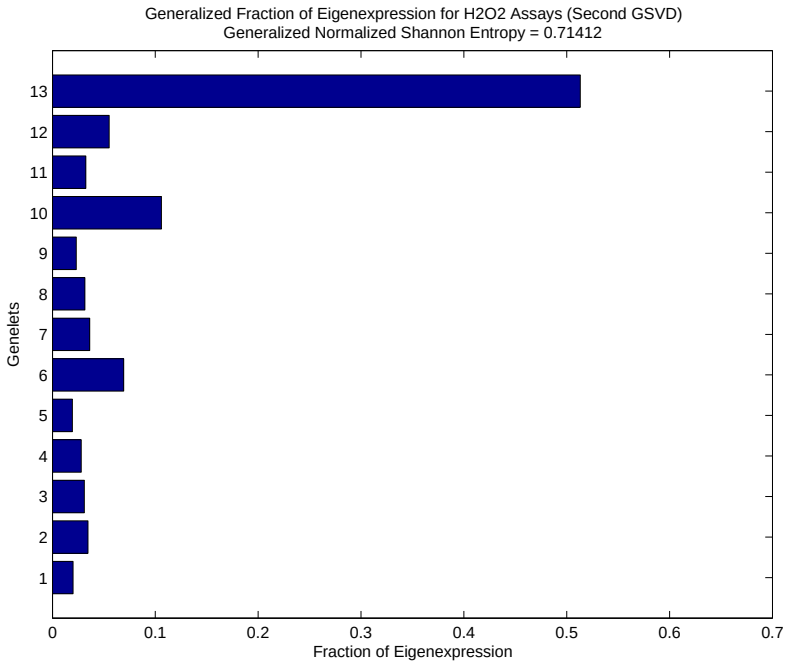


Figure 3.9: Generalized fraction of eigenexpression and normalized Shannon entropy ($D_1 = 0.71412$) for the H_2O_2 data set.

Indeed it is. Figure 3.10 displays the profiles of the 400 genes most aligned with genelet 13 and again we have the same pattern of expression as we saw in Figure 3.7. Furthermore, here we see that in the baseline data set, the oxidative stress response genes are equally expressed throughout the duration of the experiment – in the absence of the environmental stress, these genes are not cyclically regulated.

Checking the genelets for GOTerm enrichment provides yet more proof of genelet 13's identity. Table 3.2 shows that genelet 13 contains a statistically significant fraction of genes *known* to be involved in the "response to oxidative stress."

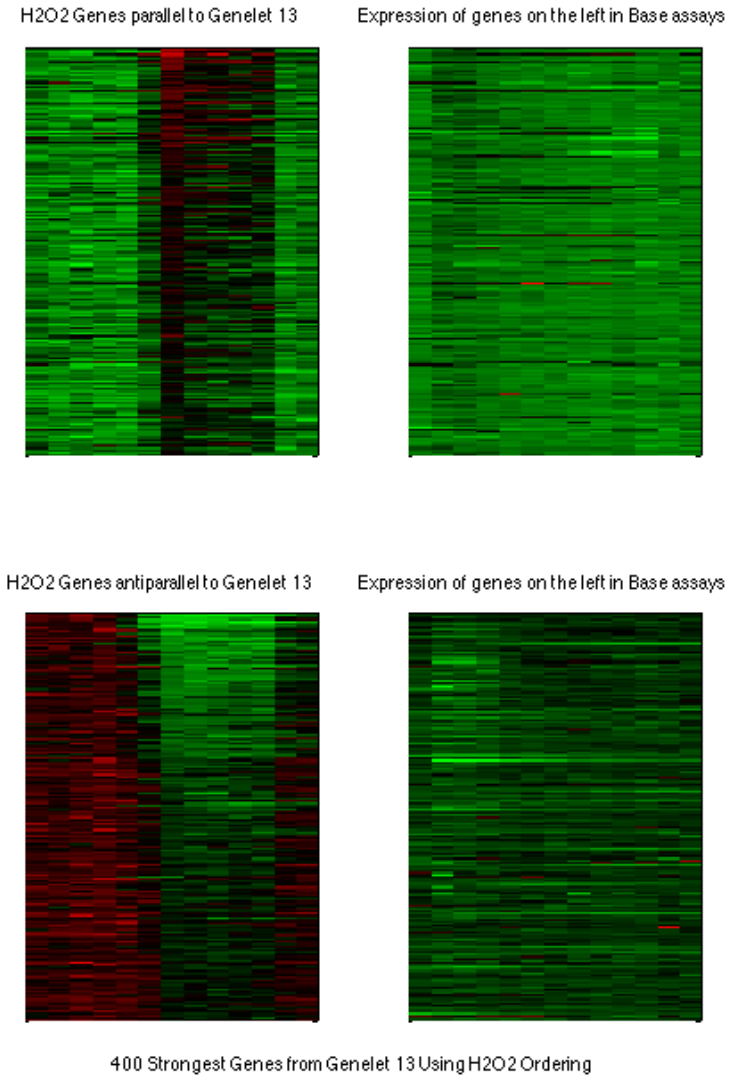


Figure 3.10: On the left, expression profiles of the 400 genes from the H_2O_2 data set with the largest (anti)parallel component along genelet 13. On the right, the expression profiles of those same genes, but in the baseline dataset. The pattern of expression in this genelet under the H_2O_2 stress is again indicative of the oxidative stress response.

Table 3.1: Statistically significant GO terms present in the H_2O_2 and MD genelets.

Genelet	Type	Ontology	Enrichment	p-value
H_2O_2 10	Process	cofactor metabolic	$25/389 = 6.4\% > 2.3\%$	0.00254
H_2O_2 10	Process	coenzyme biosynthetic	$14/389 = 3.6\% > 0.9\%$	0.00538
H_2O_2 14	Process	biogenic amine metabolic	$10/384 = 2.6\% > 0.3\%$	0.00014
H_2O_2 14	Process	amino acid derivative metabolic	$10/384 = 2.6\% > 0.4\%$	0.00112
H_2O_2 14	Process	biogenic amine biosynthetic	$8/384 = 2.1\% > 0.2\%$	0.00143
H_2O_2 14	Process	amino acid derivative biosynthetic	$8/384 = 2.1\% > 0.3\%$	0.00376
H_2O_2 14	Function	transferase activity	$62/384 = 16.1\% > 9.6\%$	0.00919
MD 9	Function	catalytic activity	$143/380 = 37.6\% > 27.4\%$	0.00376

Table 3.2: Statistically significant GO terms present in the H_2O_2 genelets (second decomposition).

Genelet	Type	Ontology	Enrichment	p-value
H_2O_2 1	Process	organelle organization and biogenesis	$114/396 = 28.8\% > 20.1\%$	0.00876
H_2O_2 8	Process	peptide metabolic process	$6/396 = 1.5\% > 0.2\%$	0.00566
H_2O_2 13	Process	response to oxidative stress	$15/393 = 3.8\% > 1.0\%$	0.00551

Chapter 4

Conclusions

The promise of GSVD is that it can discriminate between shared and unique characteristics of two data sets, allowing the separation between them to be identified and explored. As noted above and elsewhere (Gasch et al. (2000), Shapira et al. (2004)), the H_2O_2 and MD stress response profiles share many features, and thus a GSVD analysis on them was only somewhat informative.

That said, the oxidative stress response (OSR) was successfully identified in both decompositions. The yeast GO term database currently lists only 74 genes as being definitively associated with the OSR. It seems reasonable that at least some of the 400 genes displayed above as being strongly associated with the OSR genelet are also part of the OSR. Further biological experiments of the kind outlined in Section 3.2 are clearly called for to expand what is known about the yeast stress response.

4.1 Comparison with other techniques

There are at least two advantages of the GSVD-based clustering performed here over the hierarchical clustering performed in Shapira et al. (2004), one practical, the other theoretical. Practically speaking, the GSVD approach is simply much faster. The matrix decomposition required takes around one-tenth of a second on a modern

dual-core 2.4 GHz computer using `MATLAB`. By contrast, the current version of the hierarchical clustering software used in Shapira et al. (2004) is able to cluster one of the two data sets in about 28 seconds using the same computer. This difference of two orders of magnitude means that running GSVD-based clusterings to compare a new time course against many previous time courses in a brute force search for new features is a feasible proposition.

Secondly, the clusters generated by this approach are explicitly constructed to identify distinguishing features of the two time series. In the traditional hierarchical clustering approach, clusters would first have to be generated within the two data sets, and then the distinguishing clusters would have to be identified. With GSVD, this process happens in one step.

The application of GSVD to measure the separation between two matrices is useful in other machine learning contexts besides time-series feature analysis. For example, in Linear Discriminant Analysis (LDA), we wish to identify those discriminating features which maximize the "between classes scatter matrix," while minimizing the "within classes scatter matrix." This is equivalent to maximizing the separation between the two matrices. Howland and Park (2004) uses GSVD to extend LDA to cases in which the within classes scatter matrix is singular, proposing an example in which the number of terms in a document collection is much larger than the number of documents.

4.2 Future directions

Expanding on the limited biological analysis performed, such as searching the genelets for the genes which mark stages of the cell-cyle (G1, S, G2, M) and exploring the other GO term enrichments previously reported, might reveal more useful information. Additional comparison of the raw GSVD results, such as the expression levels of the

genelets over time, with the results reported by traditional techniques in the original Shapira study could further illustrate the success (or failure) of applying GSVD to this experiment. Comparing other clusters generated by the GSVD analysis with traditionally-clustered results could provide more insights like the link between Figure 3.7 and Shapira et al. (2004) Figure 3.

Appendix A

Source Code

Source code written to produce this analysis, consisting of:

1. `FirstGSVD.m` – Runs the `GSVDAnalysis` program with suitable inputs for the comparison of the Shapira et al. (2004) H_2O_2 II and MD II data sets.

2. `SecondGSVD.m` – Runs the `GSVDAnalysis` program with suitable inputs for the comparison of the Shapira et al. (2004) H_2O_2 II and Spellman et al. (1998) alpha-factor data sets.

3. `GSVDAnalysis.m` – Performs the GSVD analysis described in this paper; based upon the descriptions in Alter et al. (2003a) and Alter et al. (2003b). It calculates the antisymmetric angular distance between the components, the generalized fractions of eigenexpression, the generalized normalized Shannon entropy, produces all of the plots presented in this paper, and creates lists of the strongest genes in each genelet, which can be uploaded to the GOTermFinder.

4. `strongestGenes.m` – Helper function for `GSVDAnalysis` to display the strongest genes in a given genelet using a particular ordering (e.g., Figure 3.7).

5. `geneColormap.m` – "Heat-map" style colormap with red indicating low values and green high values.

6. `mypcolor.m` – Wrapper around the standard `MATLAB` function `pcolor`. Expands the input matrix so that the image contains a color for all rows and columns.

7. `common.py` – Python program to filter two `*.pcl` files and output two new files which contain only genes common to both inputs.

Appendix B

Expression Profiles

1. Expression profiles of the 400 genes in the H_2O_2 assays most strongly associated with genelets 1-14.

2. Expression profiles of the 400 genes in the MD assays most strongly associated with genelets 1-14.

Bibliography

Alter, O., Brown, P. O., and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, 97(18):10101–10106.

Alter, O., Brown, P. O., and Botstein, D. (2003a). Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proc. Natl. Acad. Sci. USA*, 100(6):3351–3356.

Alter, O., Brown, P. O., and Botstein, D. (2003b). Supplemental material for generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. Supplemental material – available online at http://www.pnas.org/cgi/content/full/0530258100/DC1.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Brent, R. (2000). Review: Genomic biology. *Cell*, 100(1):169–183.

Ding, C. and He, X. (2004). K-means clustering via principal component analysis. In *Proceedings of the Twenty-first International Conference on Machine Learning*, volume 69 of *ACM International Conference Proceeding Series*, Banff, Canada.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, 11(12):4241–4257.

Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computations*. Johns Hopkins University Press, 3rd edition.

Hibbs, M. A., Hess, D. C., Myers, C. L., Huttenhower, C., Li, K., and Troyanskaya, O. (2007). Exploring the functional landscape of gene expression: directed search of large microarray compendia. *Bioinformatics*, 23(20):2692–2699. The SPELL project is available online at http://avis.princeton.edu:3000/yeast/.

Howland, P. and Park, H. (2004). Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):996–1006.

Kislinger, T., Cox, B., Kannan, A., Chung, C., Hu, P., Ignatchenko, A., Scott, M. S., Gramolini, A. O., Morris, Q., Hallett, M. T., Rossant, J., Hughes, T. R., Frey, B., and Emili, A. (2006). Global survey of organ and organelle protein expression in mouse: Combined proteomic and transcriptomic profiling. *Cell*, 125(1):173–186.

Lockhart, D. J. and Winzeler, E. A. (2000). Genomics, gene expression and dna arrays. *Nature*, 405(6788):827–836.

Saldanha, A. J., Brauer, M. J., and Botstein, D. (2004). Nutritional homeostasis in batch and steady-state culture of yeast. *Mol. Biol. Cell.*, 15(9):4089–4104.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470.

Schulze, A. and Downward, J. (2000). Analysis of gene expression by microarrays: cell biologist's gold mine or minefield? *Journal of Cell Science*, 113(23):4151–4156.

Shapira, M., Segal, E., and Botstein, D. (2004). Disruption of yeast forkhead-associated cell cycle transcription by oxidative stress. *Mol. Biol. Cell*, 15(12):5659–5669.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, 9(12):3273–3297.

Wall, M. E., Dyck, P. A., and Brettin, T. S. (2001). Svdman – singular value decomposition analysis of microarray data. *Bioinformatics*, 17(6):566–568.

Whitfield, P. L., Sherlock, G., Saldanha, A. J., Murray, J. I., Ball, C. A., Alexander, K. E., Matese, J. C., Perou, C. M., Hurt, M. M., Brown, P. O., , and Botstein, D. (2002). Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Mol. Biol. Cell*, 13(6):1977–2000.

Yeung, M. K. S., Tegnér, J., and Collins, J. J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Natl. Acad. Sci. USA*, 99(9):6163–6168.

```
% Andrew Ferguson
% ORFE Independent Work
% May 13, 2008
%
% Before you begin, you must use the File -> Import Data command to load
% the tab-delimited file "Shapira04.flt.knn.avg.pcl"
%
% This creates the 4524x71 entry expression matrix (data) and the 4525x73
% matrix of column labels (contained in textdata{1,3:73}) and gene labels
% (contained in textdata{2:4525,1})

% Extract 14 time points from the H2O2_II and MD_II time series
% 0, 7, 14, 21, 28, 35, 47, 49, 56, 63, 77, 100, 120, and 140 minutes


H2O2_Values = data(1:length(data),[25:34,36:39]);p
MD_Values = data(1:length(data),[56:66,68,69,71]);

% Specify the tick marks for the Angular Distances plot
ADTick = [-pi/8 -3*pi/32 -pi/16 -pi/32 0 pi/32 pi/16 3*pi/32 pi/8];
ADTickLabel = {'-pi/8';'-3pi/32';'-pi/16';'-↵
pi/32';'0';'pi/32';'pi/16';'3pi/32';'pi/8'};

GSVDAnalysis(textdata, 'First', H2O2_Values, 'H2O2', MD_Values, 'MD',↵
ADTick, ADTickLabel);

clear ADTick;
clear ADTickLAbel;
```

```
% Andrew Ferguson
% ORFE Independent Work
% May 13, 2008
%
% Before you begin, you must use the File -> Import Data command to load
% the tab-delimited file "Shapira04.flt.knn.avg.pcl.common" as ShapiraData
%
% This creates the 4284x71 entry expression matrix (ShapiraData) and the
% 4285x73 matrix of column labels (contained in ShapiraTextdata{1,3:73})
% and gene labels (contained in ShapiraTextdata{2:4285,1})

% Then, use the File -> Import Data command to load the tab-delimited file
% "Spellman98_alphaFactor.flt.knn.avg.pcl.common" as SpellmanData
%
% This creates the 4284x17 entry expression matrix (SpellmanData) and the
% 4285x19 matrix of column labels (contained in SpellmanTextdata{1,3:19})
% and gene labels (contained in ShapiraTextdata{2:4285,1})


NumGenes = length(ShapiraData);

% Extract 13 time points from the Base (Spellman)
% 0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 98, 119 minutes
Base_Values = SpellmanData(1:NumGenes, [2:12,14,17]);

% Extract 13 time points from the H2O2_II
% 0, 7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 100, 120 minutes
H2O2_Values = ShapiraData(1:NumGenes, [25:35,37:38]);

% Specify the tick marks for the Angular Distances plot
ADTick = [-pi/8 -3*pi/32 -pi/16 -pi/32 0 pi/32 pi/16 3*pi/32 pi/8 5*pi/32↙
3*pi/16];
ADTickLabel = {'-pi/8';'-3pi/32';'-pi/16';'-↙
pi/32';'0';'pi/32';'pi/16';'3pi/32';'pi/8';'5pi/32';'3pi/16'};

GSVDAnalysis(SpellmanTextdata, 'Second', H2O2_Values, 'H2O2', Base_Values,↙
'Base', ADTick, ADTickLabel);

clear ADTick;
clear ADTickLabel;
```

```
% Andrew Ferguson
% ORFE Independent Work
% May 13, 2008
%

% Performs the GSVD analysis described in Chapter 2 - Methods on
% the matrices A and B, where aString and bString are strings which
% describe the contents of the respective matrices. gsvdString is a
% string which identifies this run of the GSVD algorithm. textdata is
% the matrix of labels for each gene.

function GSVDAnalysis(textdata, gsvdString, A, aString, B, bString, ADTick,↵
ADTickLabel)

NumStrongest = 200; % Number of strongest genes to look at in each cluster

%%
% Perform GSVD operation and normalize the genelets so they form a basis

% Produces the 'ecomony sized GSVD' where E1 and E2 are square

[U1,U2,X,E1,E2] = gsvd(A, B, 0);

X = X';         % GSVD = U * E * X' but we want to work with U * E * X

% X is now genelets x arrays   (rows x cols)
% E{1,2} is now arraylets x genelets
% U{1,2} is now genes x arraylets

NumGenelets = length(X);

% Next, normalize the genelets so that we have a set of equal-length
% basis vectors. This requires scaling the values of E{1,2} so that
% we maintain M{1,2} = U{1,2} * E{1,2} * X

normedX = zeros(NumGenelets, NumGenelets);
normedE1 = zeros(NumGenelets, NumGenelets);
normedE2 = zeros(NumGenelets, NumGenelets);

for i = 1:NumGenelets
    len = sqrt(X(i, 1:NumGenelets) * X(i, 1:NumGenelets)');

    normedX(i,1:NumGenelets) = X(i,1:NumGenelets) / len;

    normedE1(i,i) = E1(i,i) * len;
    normedE2(i,i) = E2(i,i) * len;
end

X = normedX;
E1 = normedE1;
E2 = normedE2;

%%
% Make folders to store output

[foo, bar, foobar] = mkdir([gsvdString,'GOTerms']);
```

```
clear foo; clear bar; clear foobar;

[foo, bar, foobar] = mkdir([gsvdString,'Figures']);
clear foo; clear bar; clear foobar;

FiguresFolder = [gsvdString, 'Figures/'];

figure;


%%
% Angular distances

AngularDistances = atan(diag(E1) ./ diag(E2)) - pi/4;

barh(AngularDistances, 'DisplayName', 'AngularDistances');
xlabel('Angular Distance');
ylabel('Genelets');
title({['Significance of Genelets in ',aString,' Expression Relative to ',↵
bString,' Expression'];'(Most Relatively Significant on Top)'});
set(gca, 'XTick', ADTick);
set(gca, 'XTickLabel', ADTickLabel);
xlim([min(ADTick)-0.01 max(ADTick)+0.01]);
drawnow;

saveas(gcf, [FiguresFolder,'AngularDistances.pdf'], 'pdf');

%%

genFrac(gsvdString, NumGenelets, FiguresFolder, aString, E1)
genFrac(gsvdString, NumGenelets, FiguresFolder, bString, E2)
close all hidden;

GraphicalGSVD(A, X, E1, U1)
GraphicalGSVD(B, X, E2, U2)
clusterGenelets(NumGenelets, A, U1, aString)
clusterGenelets(NumGenelets, B, U2, bString)
dumpStrongestGenes(gsvdString, textdata, FiguresFolder, NumGenelets,↵
NumStrongest, A, aString, B, bString, U1, aString)
dumpStrongestGenes(gsvdString, textdata, FiguresFolder, NumGenelets,↵
NumStrongest, A, aString, B, bString, U2, bString)


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Generalized Fractions of Eigenexpression (see Alter supporting info)
% and generalized normalized Shannon entropy
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function genFrac(gsvdString, NumGenelets, FiguresFolder, mString, E)
    GenFrac = diag(E*E) / sum(diag(E*E));
    Entropy = -1/log(NumGenelets) * GenFrac' * log(GenFrac);

    barh(GenFrac, 'DisplayName', 'GenFrac');
```

```
    xlabel('Fraction of Eigenexpression');
    ylabel('Genelets');
    title({['Generalized Fraction of Eigenexpression for ', mString, '↵
Assays (', gsvdString, ' GSVD)'], ['Generalized Normalized Shannon Entropy =↵
', num2str(Entropy)]});
    drawnow;

    saveas(gcf, [FiguresFolder,'Eigenexpression_',mString,'.pdf'], 'pdf');

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Create our version of Alter's "Figure 5", which is a graphical display
% of the GSVD.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function GraphicalGSVD(M, X, E, U)
    figure;
    set(gcf, 'Position', [227 2 866 678]);
    colormap(geneColormap);

    % Start with the actual expression levels

    frame = subplot(3, 4, [1 5 9]);
    h = mypcolor(M);
    set(h, 'LineStyle', 'none');
    set(frame, 'XAxisLocation', 'top');
    set(frame, 'YDir', 'reverse');
    set(frame, 'YTickLabel', {});
    xlabel(frame, 'Arrays');
    ylabel(frame, 'Genes');

    % X -- Arrays x Genelets

    frame = subplot(3, 4, 4);
    h = mypcolor(X);
    set(h, 'LineStyle', 'none');
    set(frame, 'XAxisLocation', 'top');
    set(frame, 'YDir', 'reverse');
    xlabel(frame, 'Arrays (Time Periods)');
    ylabel(frame, 'Genelets');

    % E -- Genelets x Arraylets

    frame = subplot(3, 4, 3);
    h = mypcolor(E);
    set(h, 'LineStyle', 'none');
    set(frame, 'XAxisLocation', 'top');
    set(frame, 'YDir', 'reverse');
    xlabel(frame, 'Genelets');
    ylabel(frame, 'Arraylets');

    % U -- Genes x Arraylets

    frame = subplot(3, 4, [2 6 10]);
```

```
    h = mypcolor(U);
    set(h, 'LineStyle', 'none');
    set(frame, 'XAxisLocation', 'top');
    set(frame, 'YDir', 'reverse');
    set(frame, 'YTickLabel', {});
    xlabel(frame, 'Arraylets');
    ylabel(frame, 'Genes');

    drawnow;


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Clustering of gene profiles by genelet which they are most strongly
% associated with
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function clusterGenelets(NumGenelets, Values, U, mString)
    figure;
    set(gcf, 'Position', [227 2 120 633]);
    set(gcf, 'PaperPositionMode', 'auto');

    [foo I] = max(U, [], 2);
    [foo Iabs] = max(abs(U), [], 2);

    Genelet_Assignments = Iabs+(isnan((I-Iabs) ./ (I-Iabs))*NumGenelets);

    [foo Sorted_by_Genelet] = sort(Genelet_Assignments);
    clear foo;

    colormap(geneColormap);
    h = mypcolor(Values(Sorted_by_Genelet, 1:NumGenelets));
    set(h, 'LineStyle', 'none');
    set(gca, 'XAxisLocation', 'top');
    set(gca, 'YDir', 'reverse');
    set(gca, 'YTickLabel', {});
    xlabel(gca, 'Arrays');
    ylabel(gca, 'Genes');
    title({[mString, ' Clustering']});

    drawnow;

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Now we want to identify the strongest genes for each genelet. The value
% of U{1,2}(i,j) indicates the project of the ith gene onto the jth
% genetlet. Thus, the larger (in magnitude) the value of U{1,2}(i, j) the
% stronger gene i is in genelet j.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function dumpStrongestGenes(gsvdString, textdata, FiguresFolder,↙
NumGenelets, NumStrongest, A, aString, B, bString, U, mString)
```

```
    NumGenes = length(A);
    [foo Orderings] = sort(U);
    clear foo

    % Let's look at the expression patterns of the actual genes
    % that are the strongest components in each genelet

    mkdir([FiguresFolder,mString,'_Ordering']);

    for i = 1:NumGenelets,
        strongestGenes(NumGenelets, i, NumStrongest, Orderings, [mString, '↙
Ordering'], A, aString, B, bString);
        drawnow;
    end

    % Dump files to put into Gene Ontology Finder
    % http://db.yeastgenome.org/cgi-bin/GO/goTermFinder

    cd([gsvdString, 'GOTerms']);

    for i=1:NumGenelets,
        dlmwrite(['GOTerms-',mString,'-',num2str(i),'.txt'], char([textdata↙
(Orderings(1:NumStrongest,i),1); textdata(Orderings(NumGenes-NumStrongest+1:↙
NumGenes,i),1)]), '');
    end

    cd('..')
```

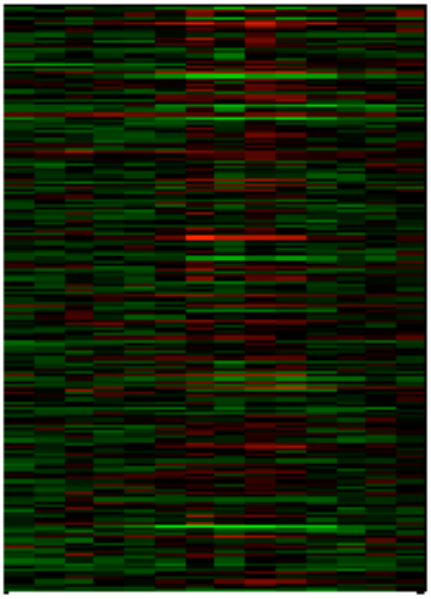
```
% Andrew Ferguson
% ORFE Independent Work
% May 13, 2008
%

% Display the N most parallel and N most antiparallel genelets to the
% specified genelet. Use the ordering provided by the specified order
% array, and select NumGenelets columns from the specified A and B
% matrices.

function strongestGenes(NumGenelets, genelet, N, order, orderStr, A, aStr,↵
B, bStr)
    totalGenes = length(A);
    fig = figure;
    set(fig, 'Position', [227 2 488 669]);
    colormap(geneColormap);

    frame = subplot(2, 2, 1);
    h = mypcolor(A(order(1:N,genelet),1:NumGenelets));
    title(frame, [aStr,' Genes parallel to Genelet ',num2str(genelet)])
    set(h, 'LineStyle', 'none');
    set(frame, 'YDir', 'reverse');
    set(frame, 'XTickLabel', {});
    set(frame, 'YTickLabel', {});
    set(frame, 'XAxisLocation', 'top');

    frame = subplot(2, 2, 3);
    h = mypcolor(A(order(totalGenes-N+1:totalGenes,genelet),1:NumGenelets));
    title(frame, [aStr,' Genes antiparallel to Genelet ',num2str(genelet)])
    set(h, 'LineStyle', 'none');
    set(frame, 'XTickLabel', {});
    set(frame, 'YTickLabel', {});
    set(frame, 'XAxisLocation', 'top');

    frame = subplot(2, 2, 2);
    h = mypcolor(B(order(1:N,genelet),1:NumGenelets));
    title(frame, ['Expression of genes on the left in ',bStr,' assays'])
    set(h, 'LineStyle', 'none');
    set(frame, 'YDir', 'reverse');
    set(frame, 'XTickLabel', {});
    set(frame, 'YTickLabel', {});
    set(frame, 'XAxisLocation', 'top');

    frame = subplot(2, 2, 4);
    h = mypcolor(B(order(totalGenes-N+1:totalGenes,genelet),1:NumGenelets));
    title(frame, ['Expression of genes on the left in ',bStr,' assays'])
    set(h, 'LineStyle', 'none');
    set(frame, 'XTickLabel', {});
    set(frame, 'YTickLabel', {});
    set(frame, 'XAxisLocation', 'top');

    text(-1, -20, [num2str(2*N), ' Strongest Genes from Genelet ', num2str↵
(genelet), ' Using ', orderStr], 'Interpreter' , 'none',↵
'HorizontalAlignment', 'center')
```

```
% Andrew Ferguson
% ORFE Independent Work
% May 13, 2008
%
% This is based off an example on The Mathworks' website, home of MATLAB

function map=geneColormap
    % Red-Green Color Map

    NumColors = 64;
    scale = [0.5: -1/(NumColors-1):-0.5];
    scale = abs(scale).*2;

    Red   = [scale(:, 1:NumColors/2) zeros(1,NumColors/2)];
    Green = [zeros(1, NumColors/2) scale(:,(NumColors/2 + 1):NumColors)];
    Blue  = zeros(1,NumColors);

    map=[Red' Green' Blue'];
```

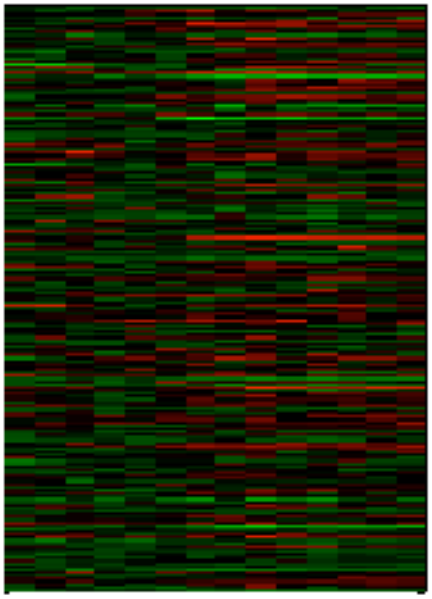
```
% Andrew Ferguson
% ORFE Independent Work
% May 13, 2008
%

% MATLAB's standard pcolor() function drops the last row and column,
% so we add on a dummy row and column before calling the function

function h=mypcolor(m)
    rows = size(m,1);
    cols = size(m,2);

    extra_col = zeros(rows, 1);
    extra_row = zeros(1, cols+1);

    h = pcolor(vertcat(horzcat(m, extra_col), extra_row));
```

```
#!/usr/bin/env python

# Andrew Ferguson
# ORFE Independent Work
# May 13, 2008
#
# Takes two tab-delimited gene expression files and processes
# them so that only genes common to both files remain.
#

import sys

def main():
        try:
                first = open(sys.argv[1])
                second = open(sys.argv[2])
        except:
                print "common.py first_file second_file"
                return

        cleanedFirst = open(sys.argv[1] + '.common', 'w+')
        cleanedSecond = open(sys.argv[2] + '.common', 'w+')

        # preserve 1 line of header information
        cleanedFirst.write(first.readline())
        cleanedSecond.write(second.readline())

        firstDict = file2dict(first)
        secondDict = file2dict(second)

        for key in firstDict.iterkeys():
                if secondDict.has_key(key):
                        cleanedFirst.write(key + "\t" + firstDict[key])
                        cleanedSecond.write(key + "\t" + secondDict[key])

        cleanedFirst.close()
        cleanedSecond.close()

def file2dict(file):
        levels = {}
        line = file.readline()

        while (line != ""):
                (gene, level) = line.split(None, 1)
                levels[gene] = level
                line = file.readline()

        return levels

if __name__ == '__main__':
        main()
```

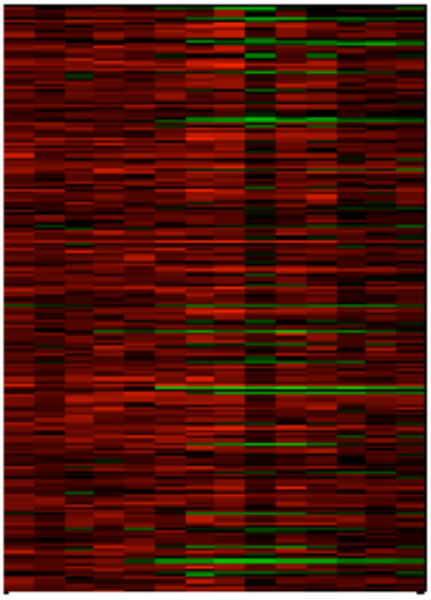
H2O2 Genes parallel to Genelet 1



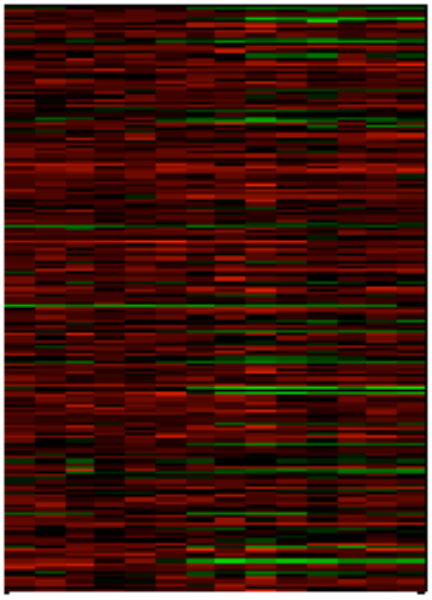
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 1

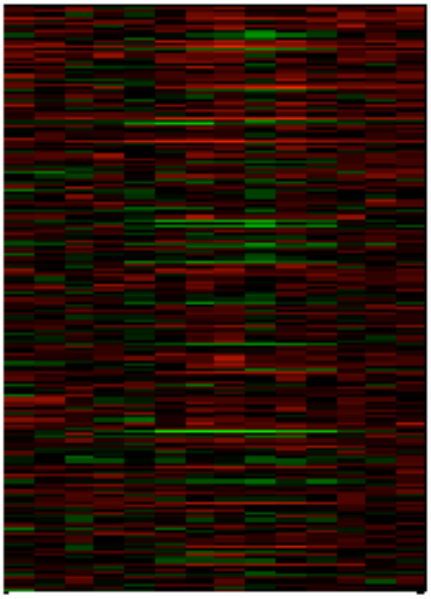


Expression of genes on the left in MD assays

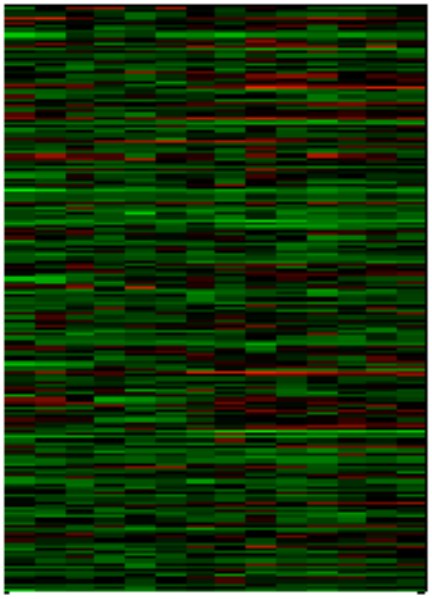


400 Strongest Genes from Genelet 1 Using H2O2 Ordering

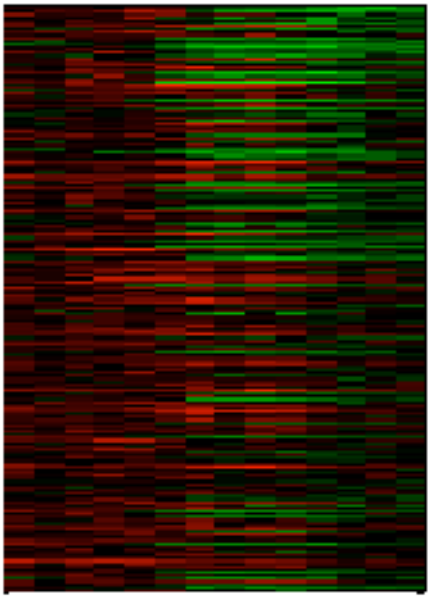
H2O2 Genes parallel to Genelet 2



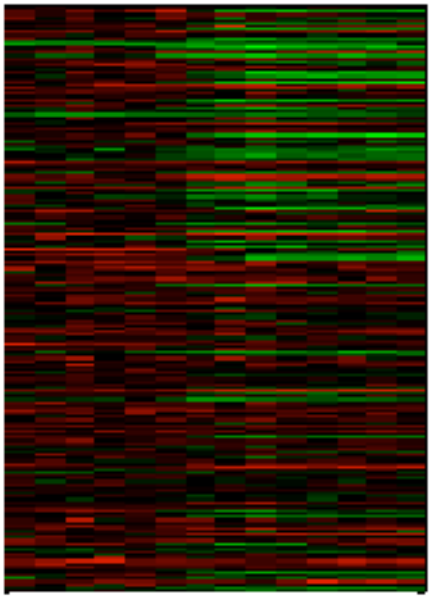
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 2

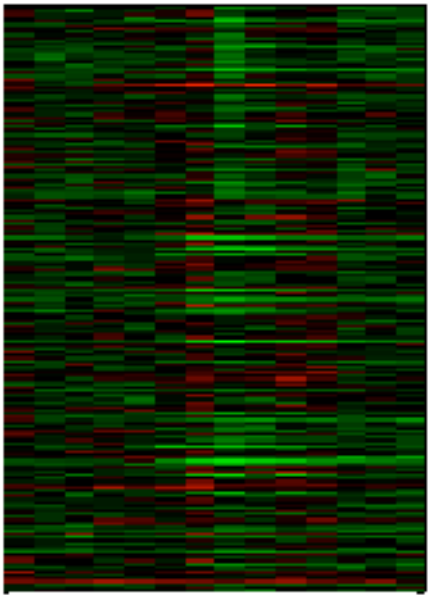


Expression of genes on the left in MD assays

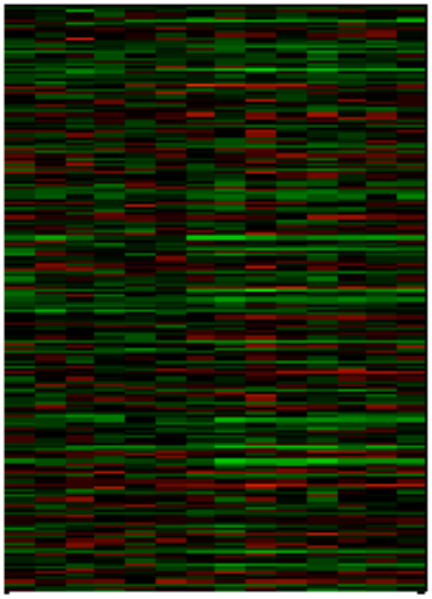


400 Strongest Genes from Genelet 2 Using H2O2 Ordering

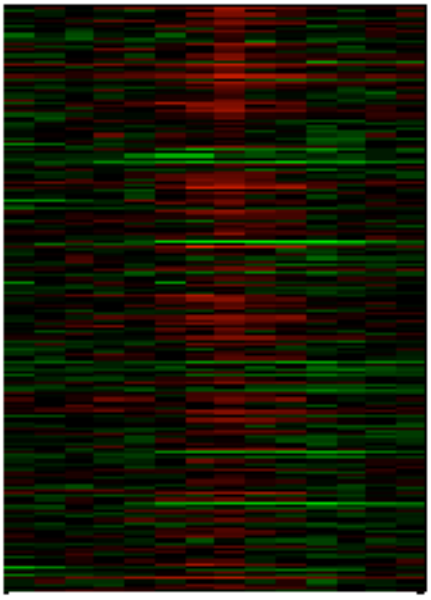
H2O2 Genes parallel to Genelet 3



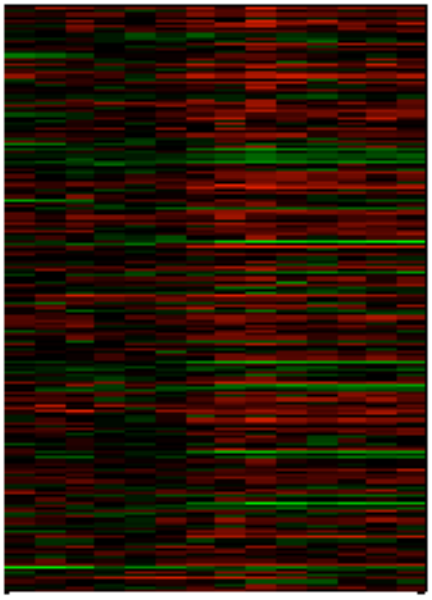
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 3

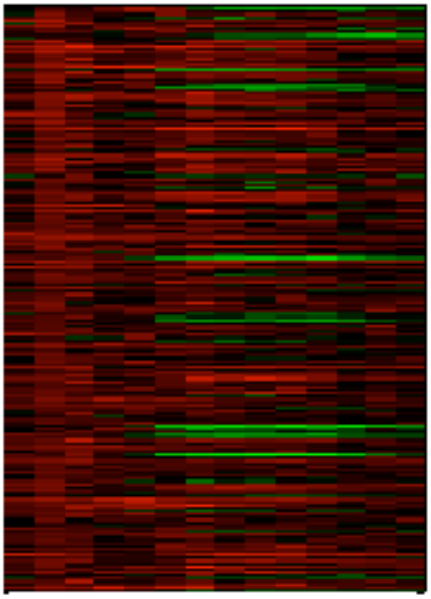


Expression of genes on the left in MD assays

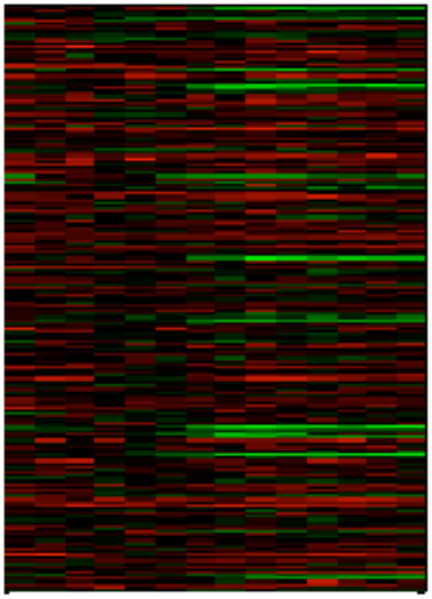


400 Strongest Genes from Genelet 3 Using H2O2 Ordering

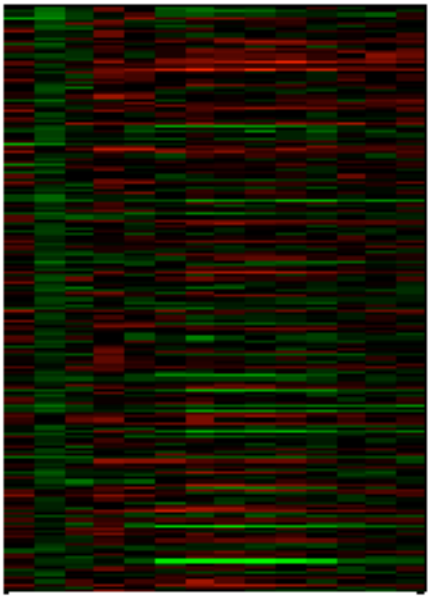
H2O2 Genes parallel to Genelet 4



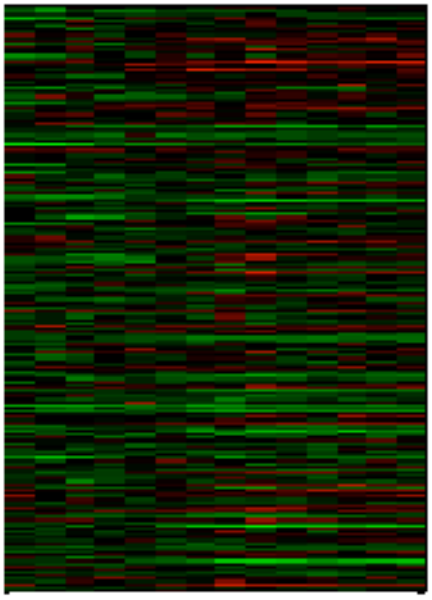
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 4

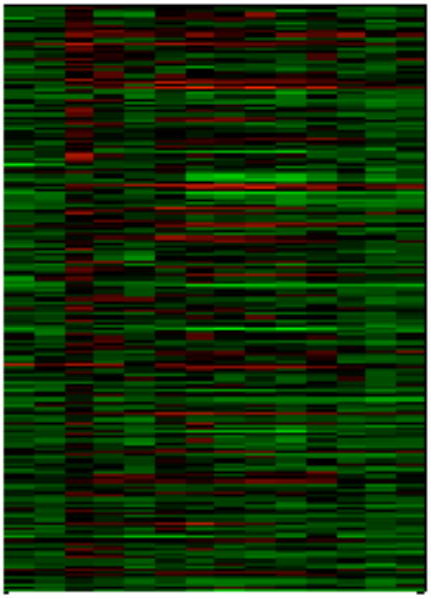


Expression of genes on the left in MD assays

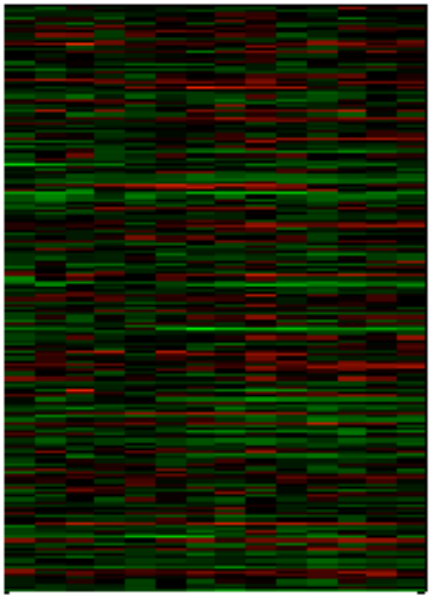


400 Strongest Genes from Genelet 4 Using H2O2 Ordering

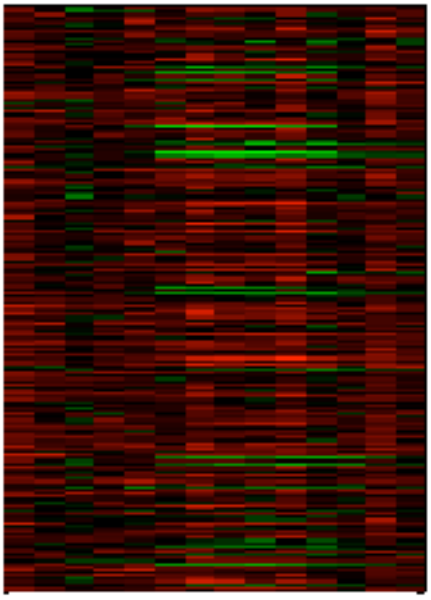
H2O2 Genes parallel to Genelet 5



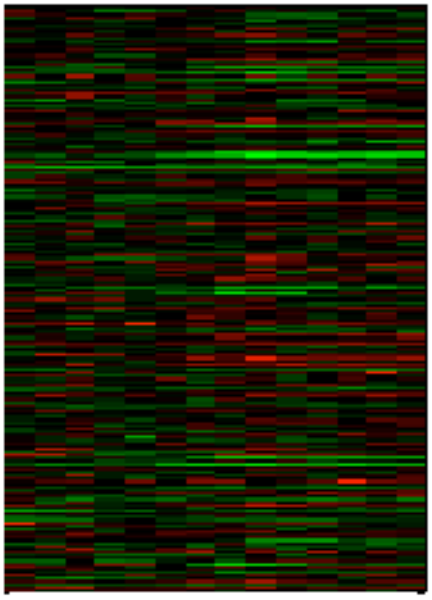
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 5

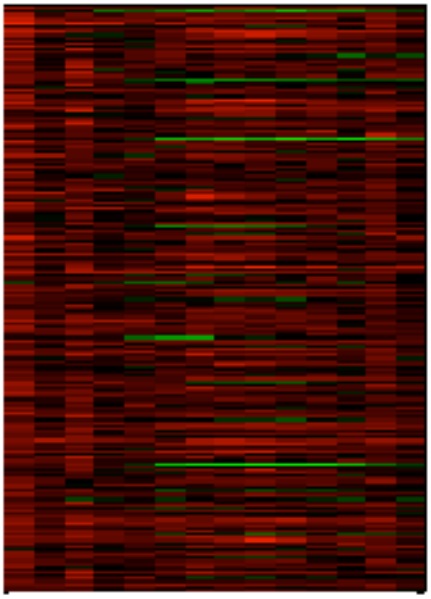


Expression of genes on the left in MD assays

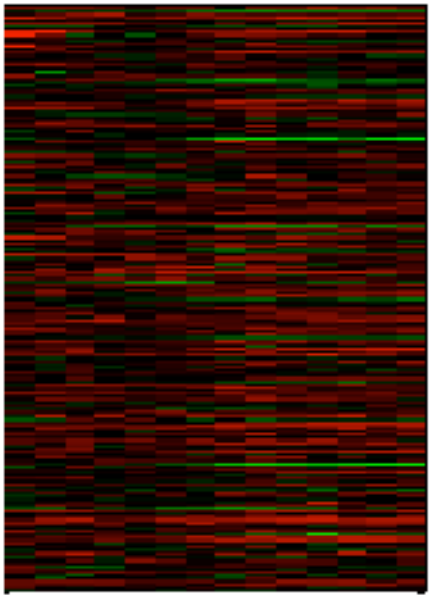


400 Strongest Genes from Genelet 5 Using H2O2 Ordering

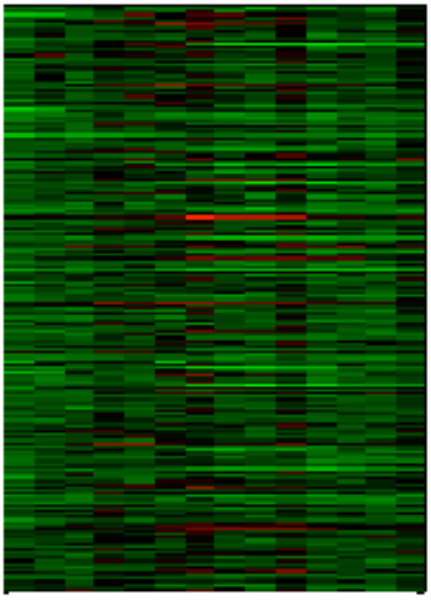
H2O2 Genes parallel to Genelet 6



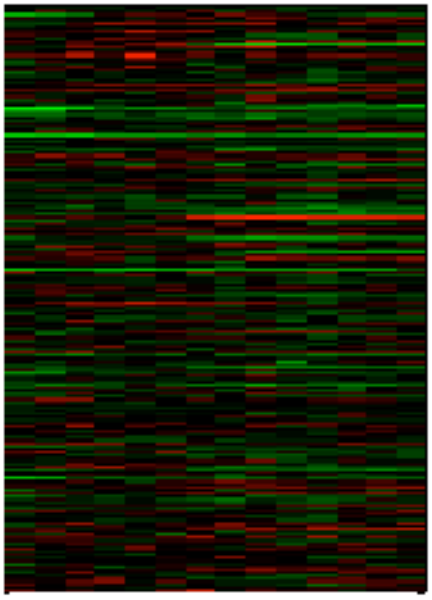
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 6

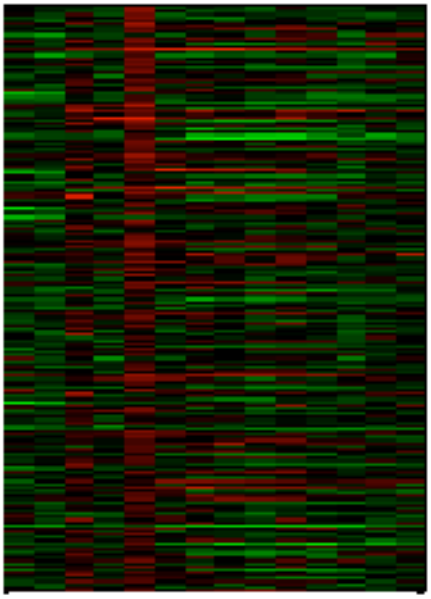


Expression of genes on the left in MD assays

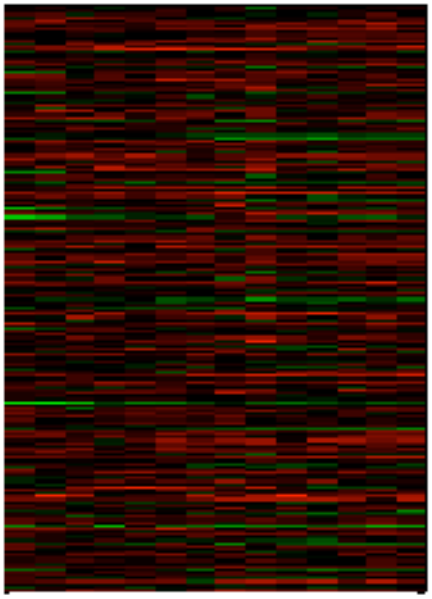


400 Strongest Genes from Genelet 6 Using H2O2 Ordering

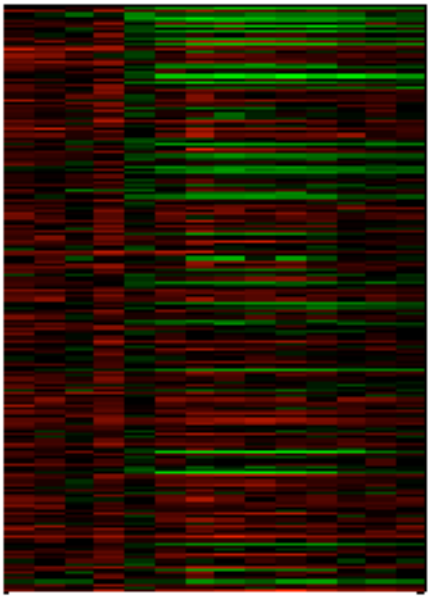
H2O2 Genes parallel to Genelet 7



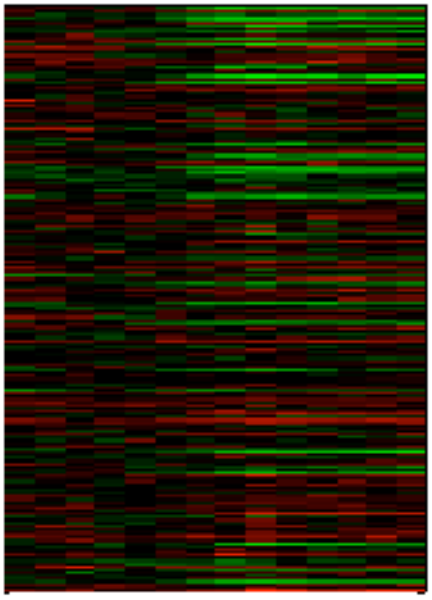
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 7

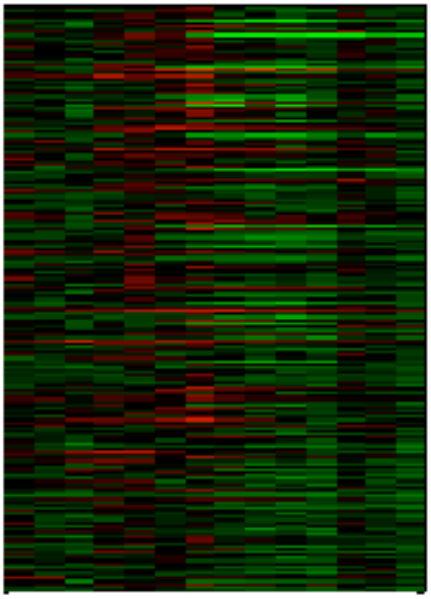


Expression of genes on the left in MD assays

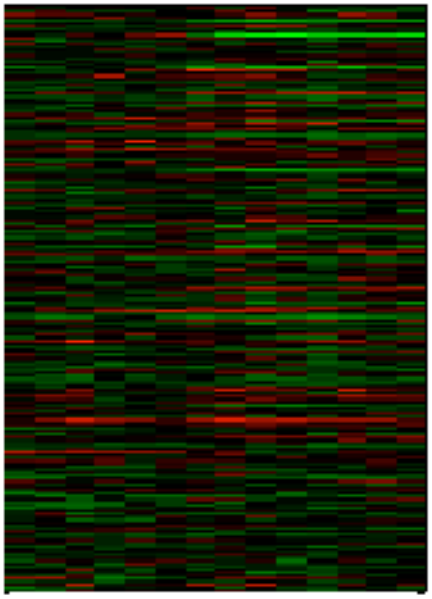


400 Strongest Genes from Genelet 7 Using H2O2 Ordering

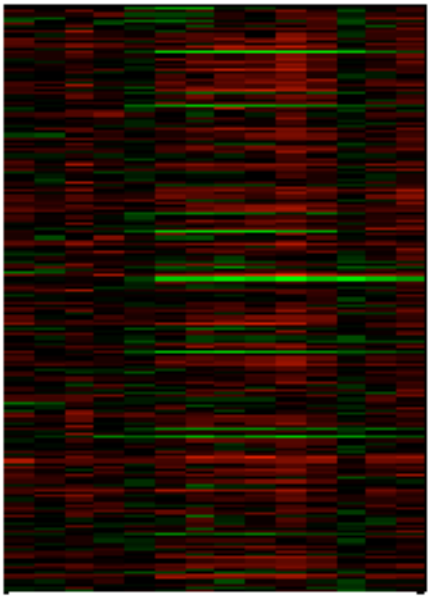
H2O2 Genes parallel to Genelet 8



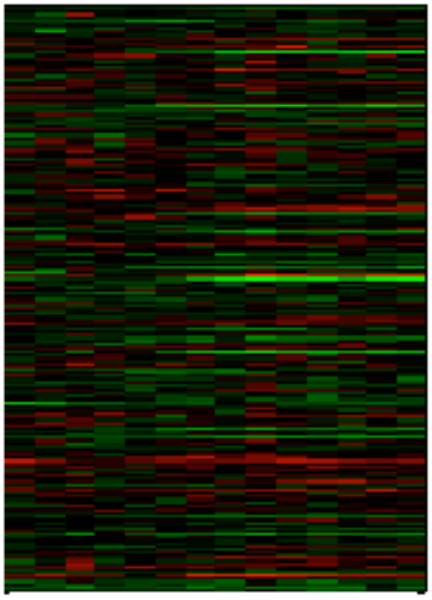
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 8

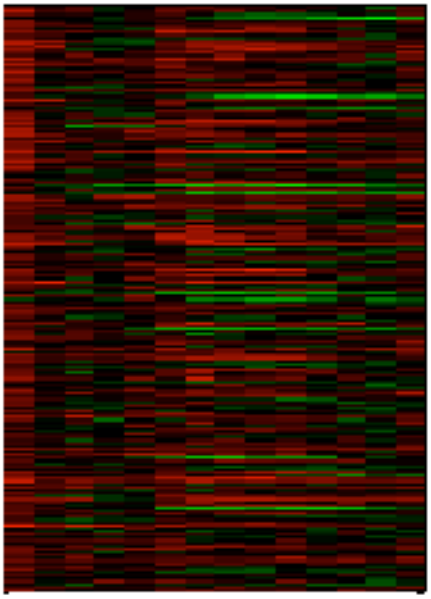


Expression of genes on the left in MD assays

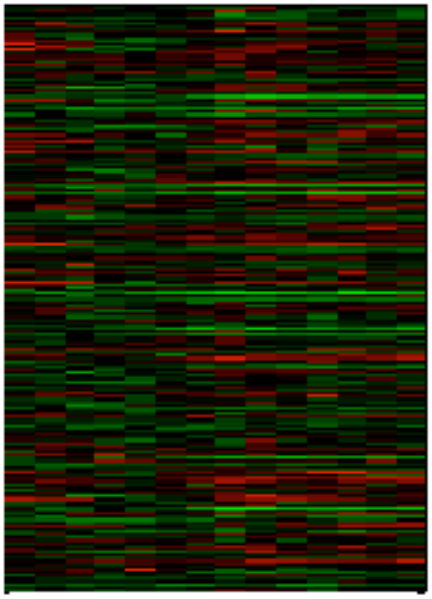


400 Strongest Genes from Genelet 8 Using H2O2 Ordering

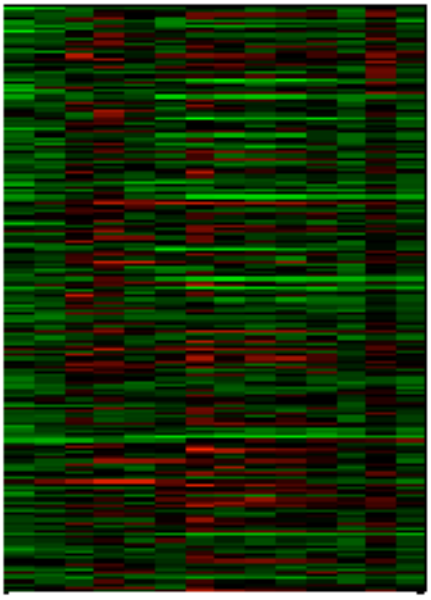
H2O2 Genes parallel to Genelet 9



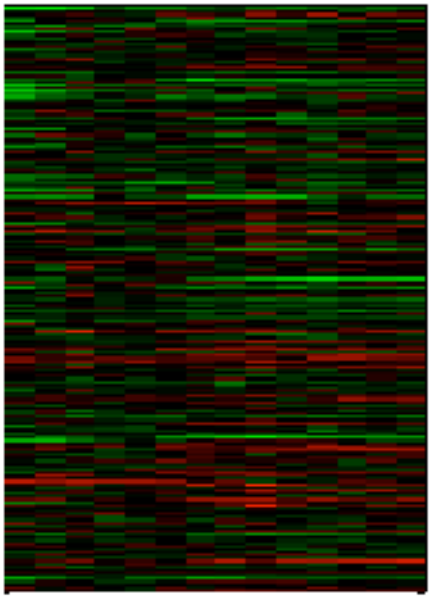
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 9

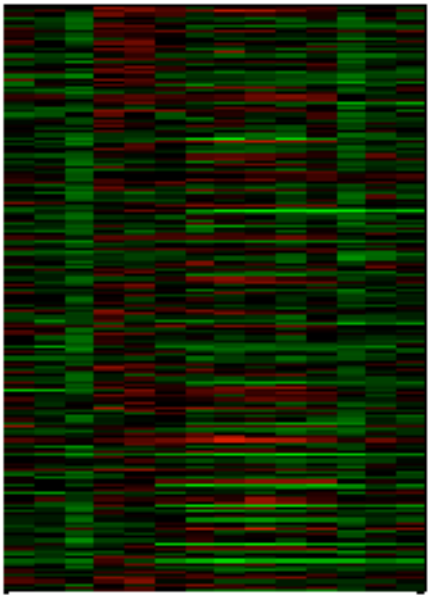


Expression of genes on the left in MD assays

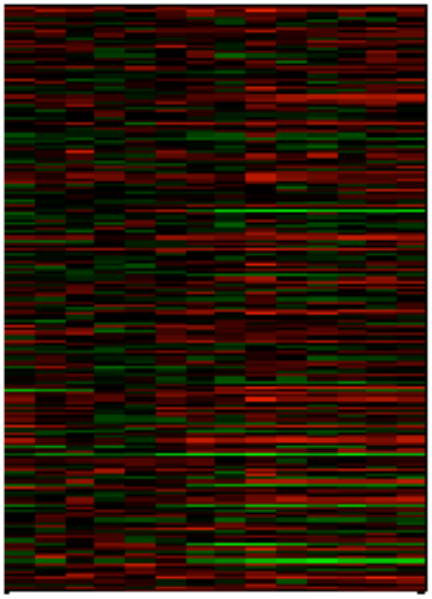


400 Strongest Genes from Genelet 9 Using H2O2 Ordering

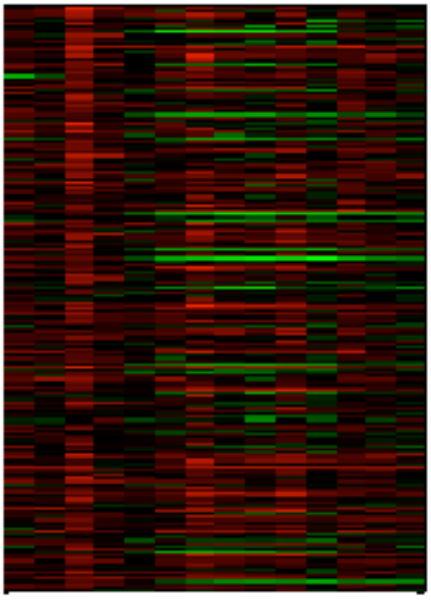
H2O2 Genes parallel to Genelet 10



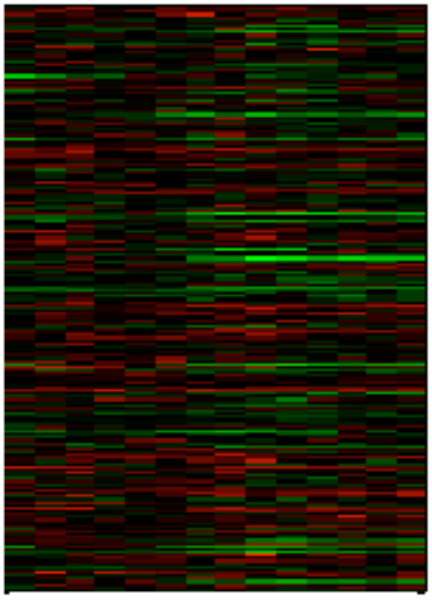
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 10

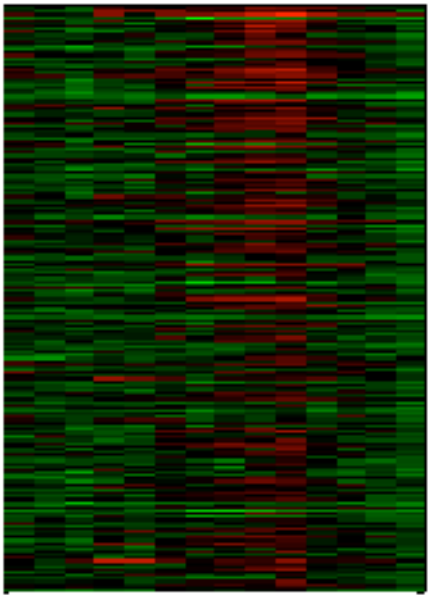


Expression of genes on the left in MD assays

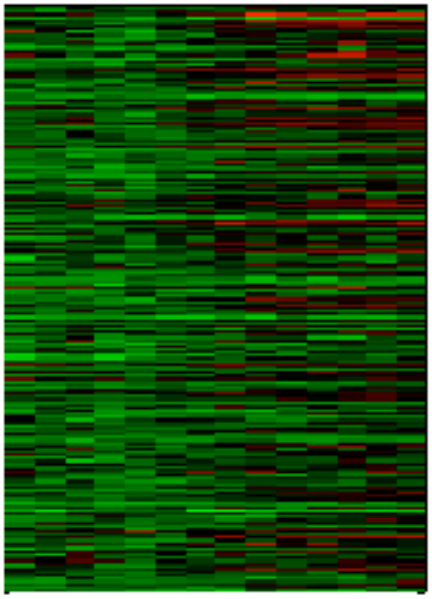


400 Strongest Genes from Genelet 10 Using H2O2 Ordering

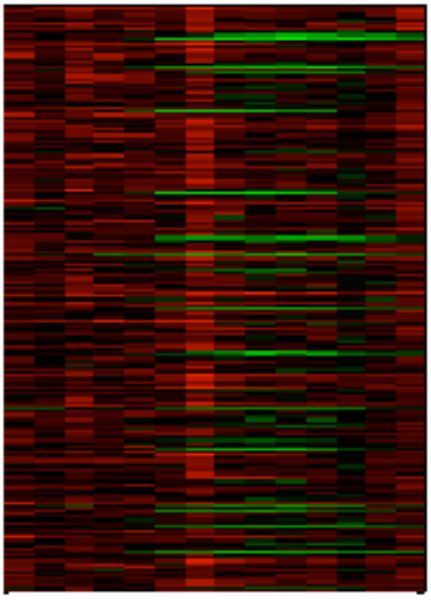
H2O2 Genes parallel to Genelet 11



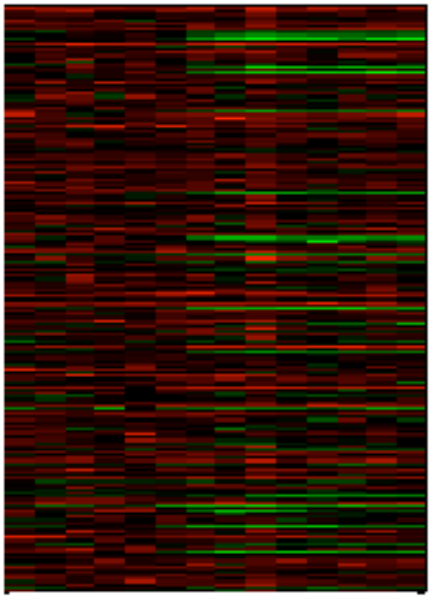
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 11

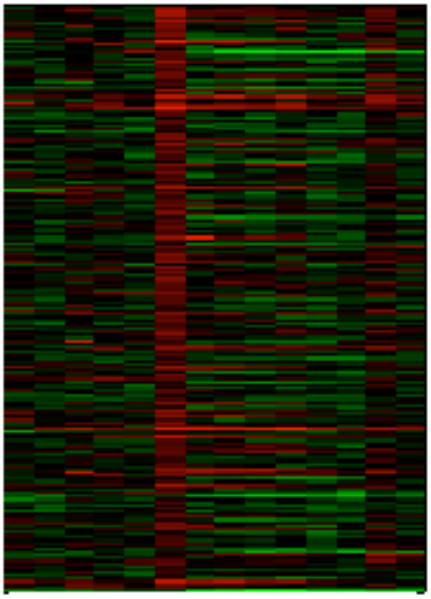


Expression of genes on the left in MD assays

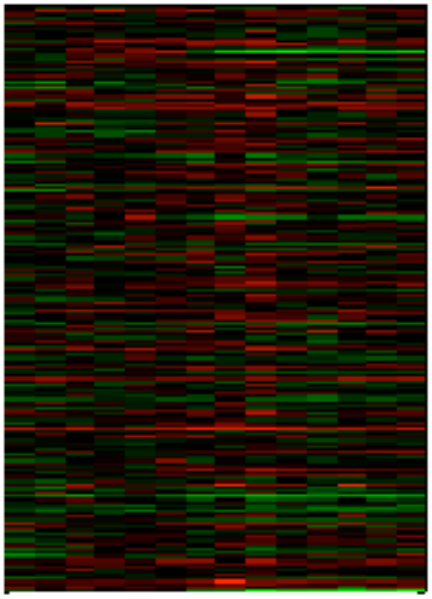


400 Strongest Genes from Genelet 11 Using H2O2 Ordering

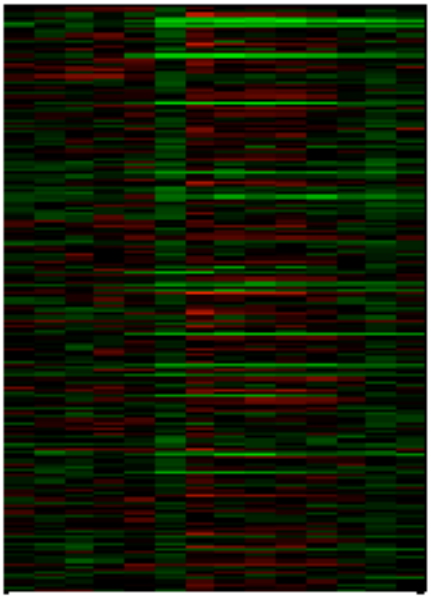
H2O2 Genes parallel to Genelet 12



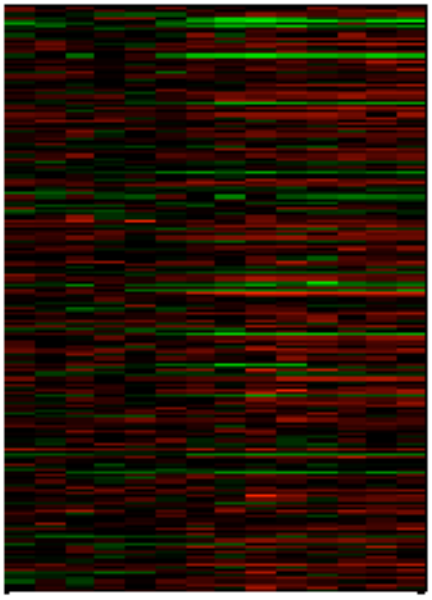
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 12

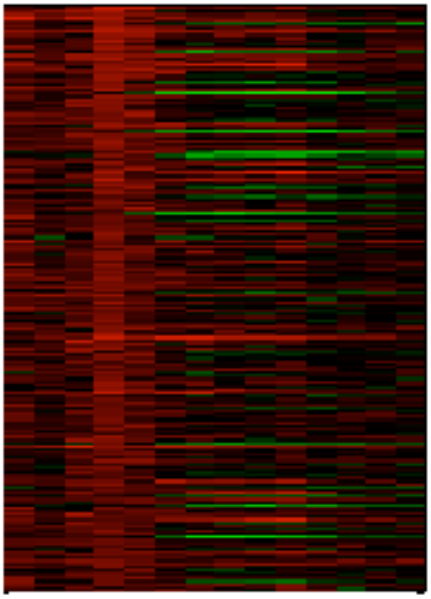


Expression of genes on the left in MD assays

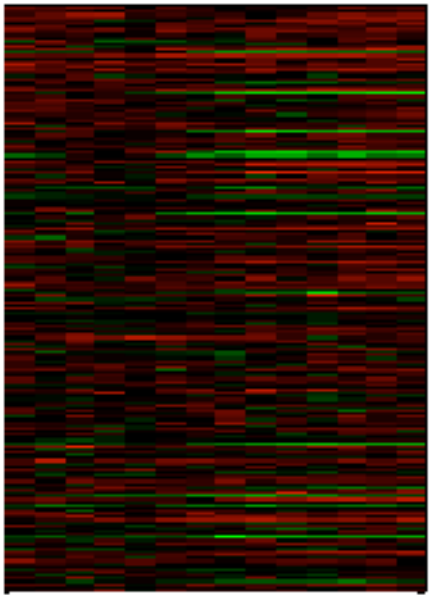


400 Strongest Genes from Genelet 12 Using H2O2 Ordering

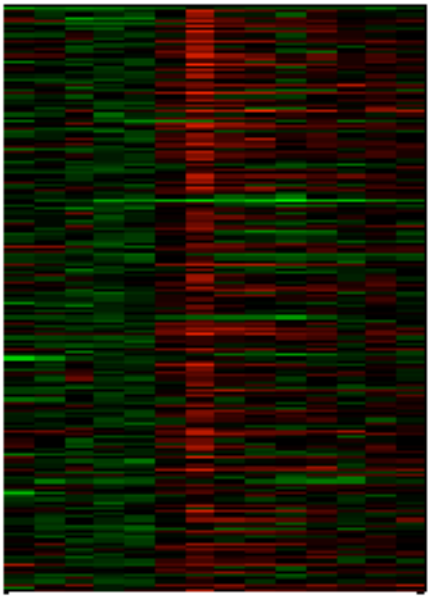
H2O2 Genes parallel to Genelet 13



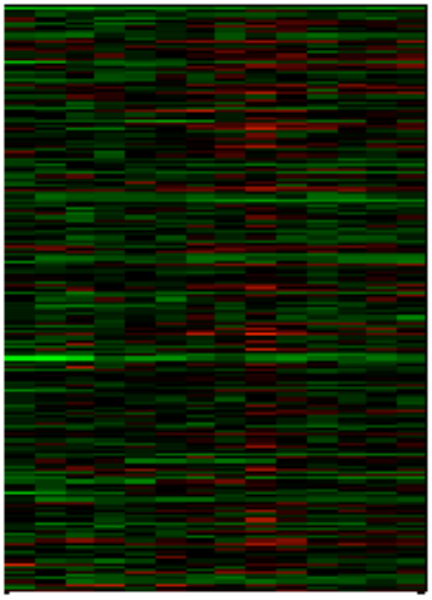
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 13

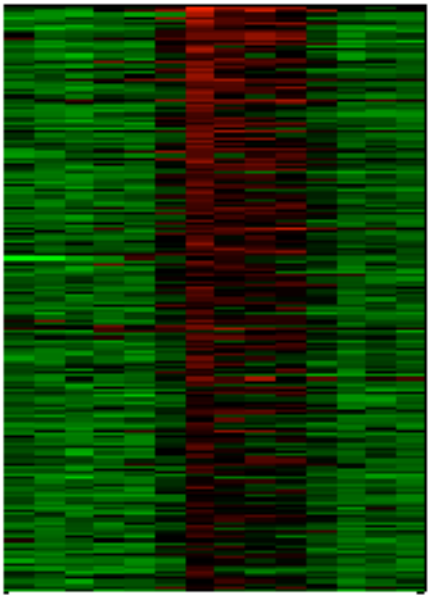


Expression of genes on the left in MD assays

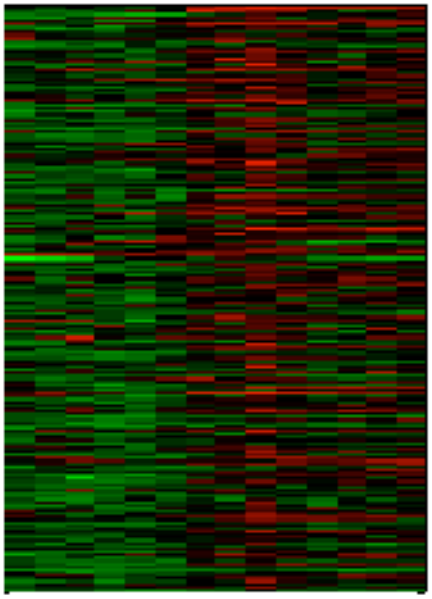


400 Strongest Genes from Genelet 13 Using H2O2 Ordering

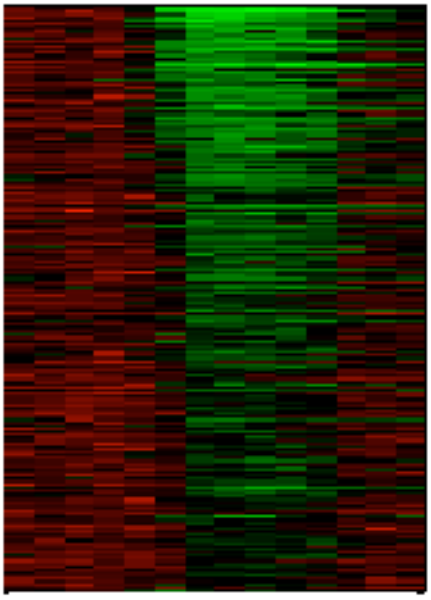
H2O2 Genes parallel to Genelet 14



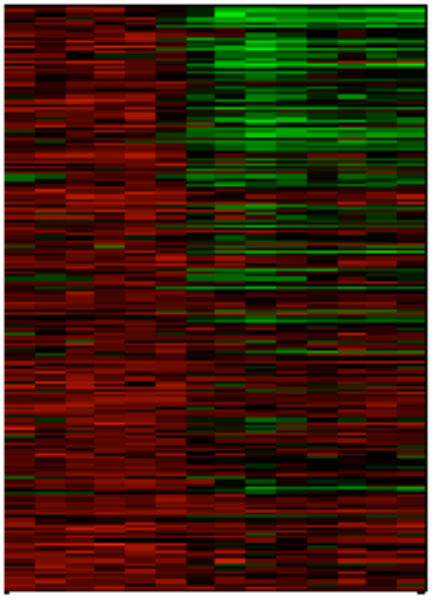
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 14

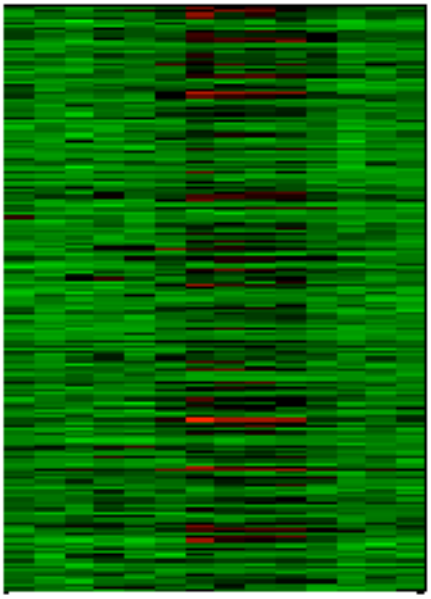


Expression of genes on the left in MD assays

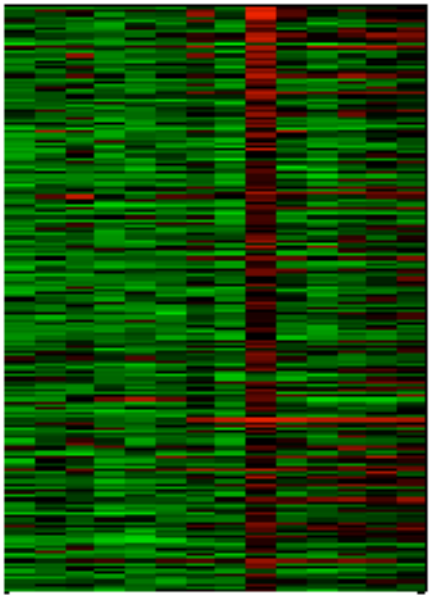


400 Strongest Genes from Genelet 14 Using H2O2 Ordering

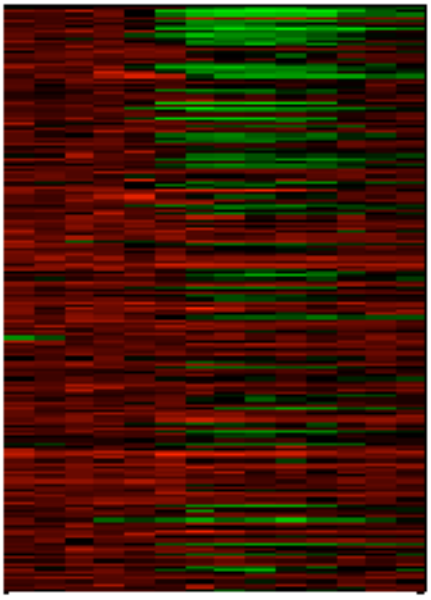
H2O2 Genes parallel to Genelet 1



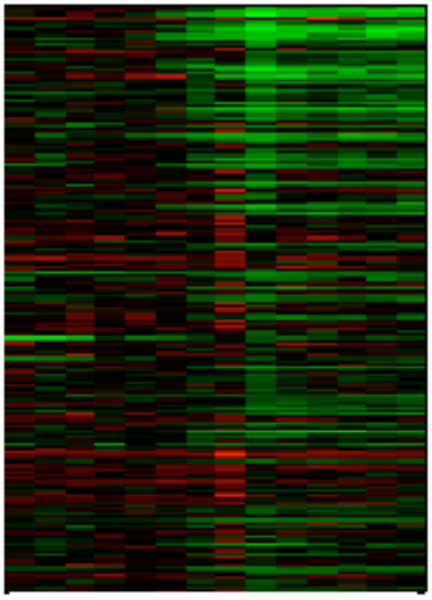
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 1

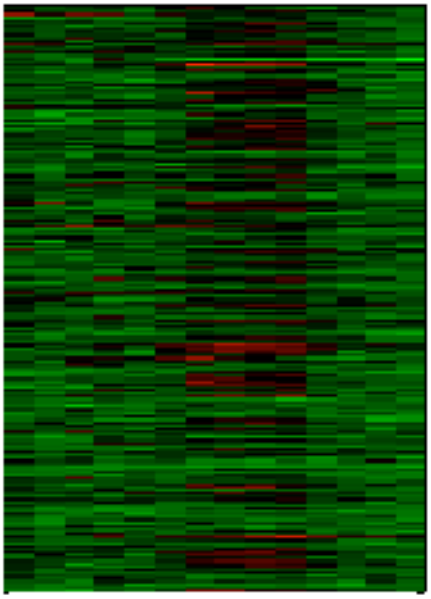


Expression of genes on the left in MD assays

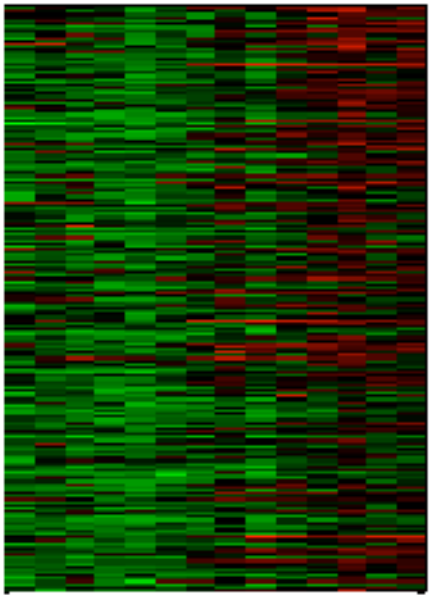


400 Strongest Genes from Genelet 1 Using MD Ordering

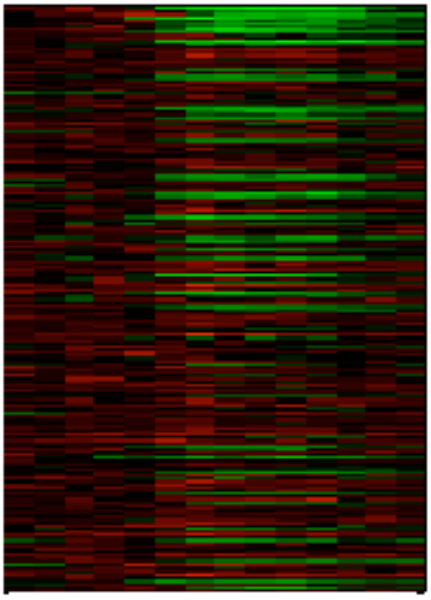
H2O2 Genes parallel to Genelet 2



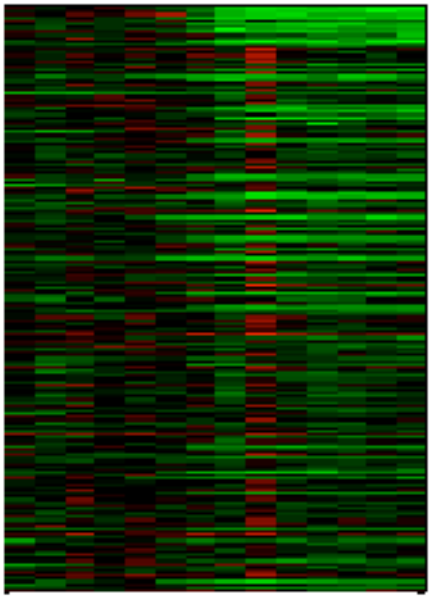
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 2

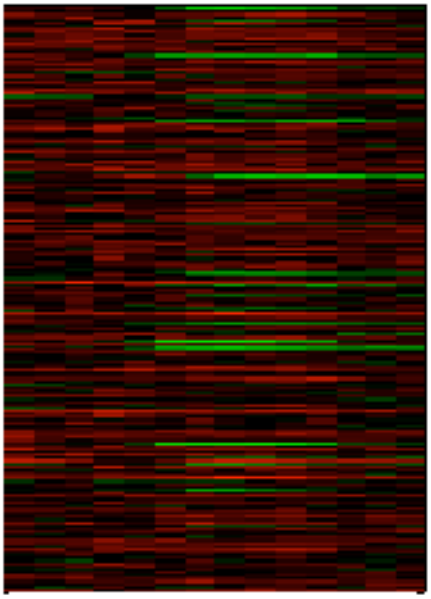


Expression of genes on the left in MD assays

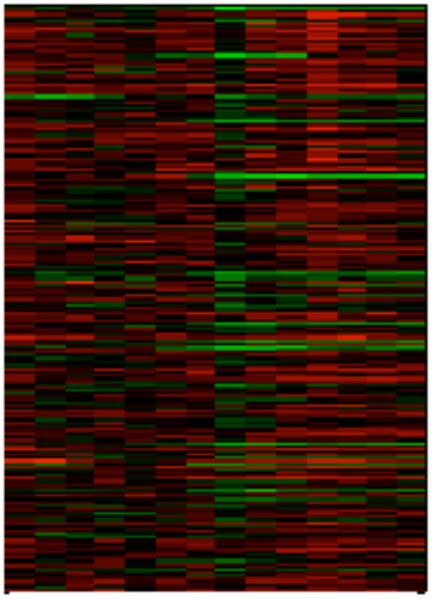


400 Strongest Genes from Genelet 2 Using MD Ordering

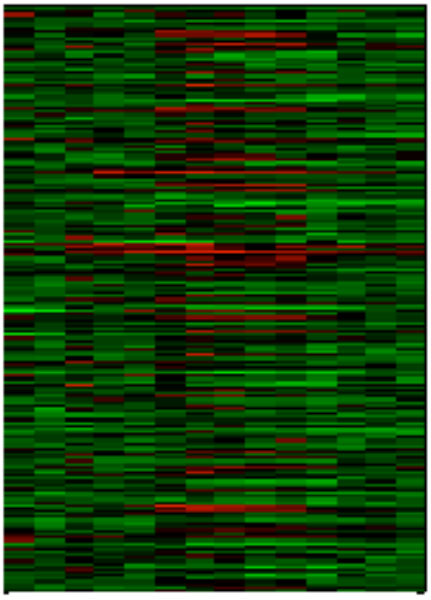
H2O2 Genes parallel to Genelet 3



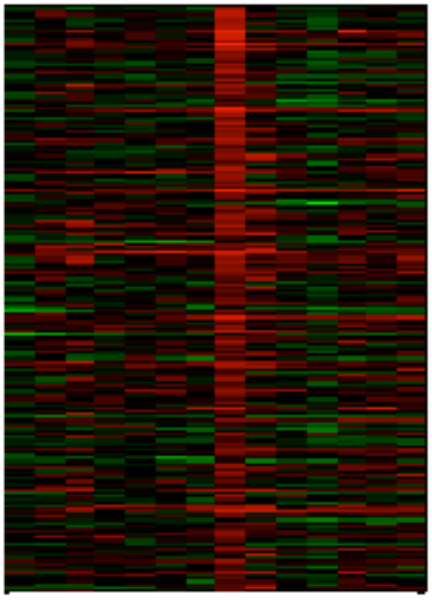
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 3

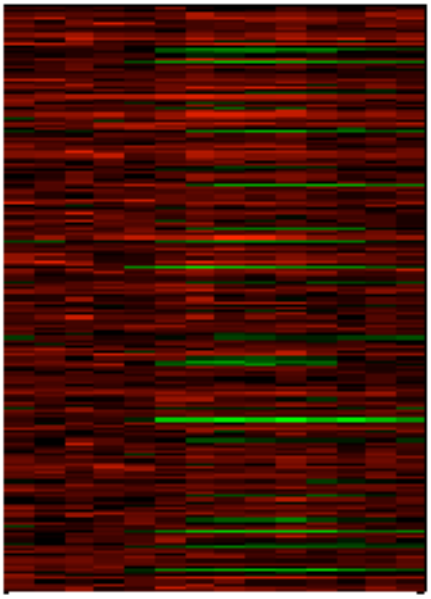


Expression of genes on the left in MD assays

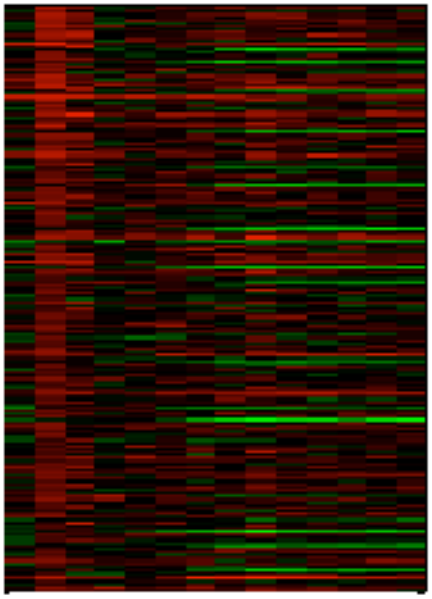


400 Strongest Genes from Genelet 3 Using MD Ordering

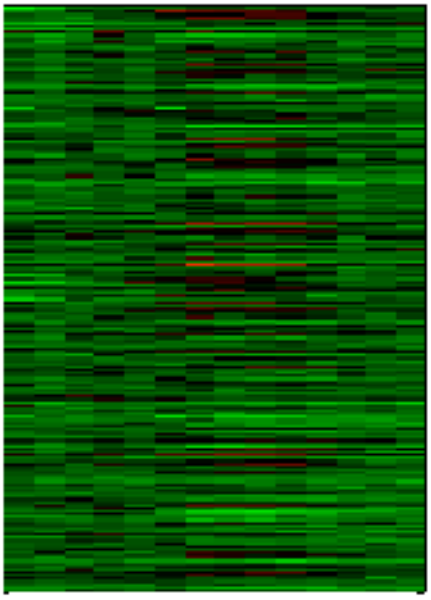
H2O2 Genes parallel to Genelet 4



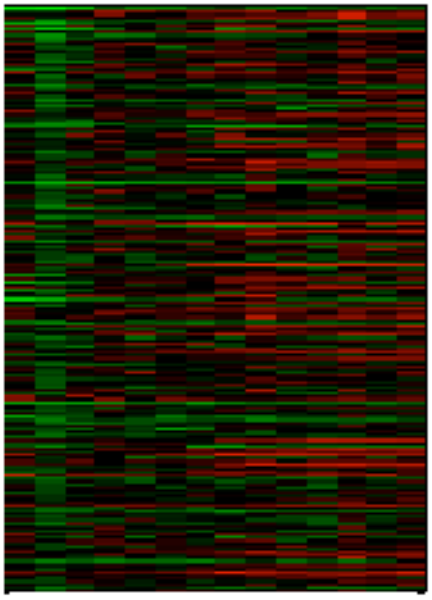
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 4

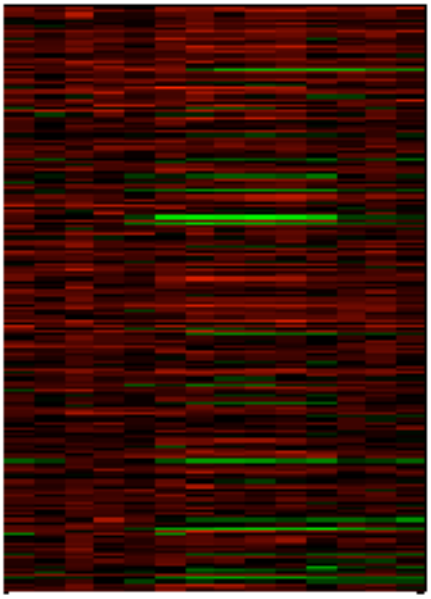


Expression of genes on the left in MD assays

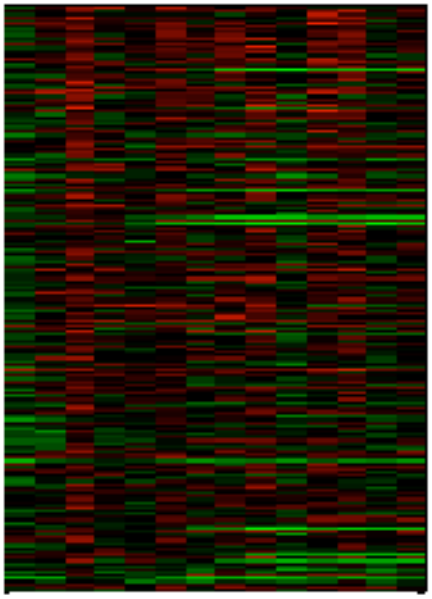


400 Strongest Genes from Genelet 4 Using MD Ordering

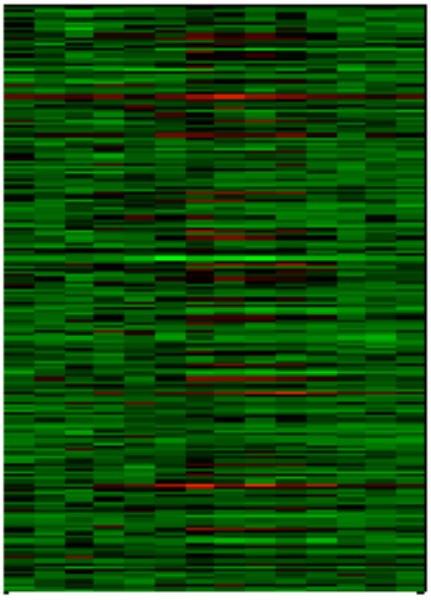
H2O2 Genes parallel to Genelet 5



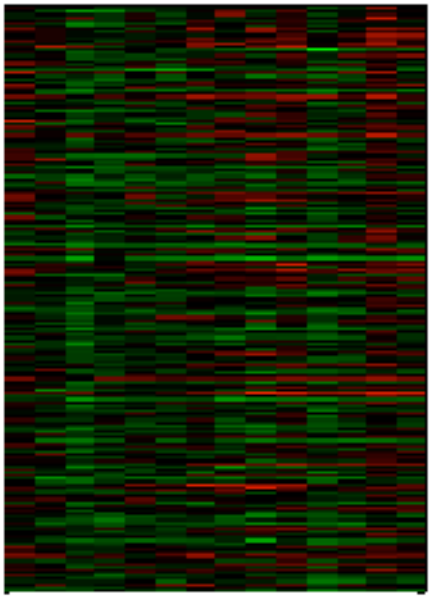
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 5

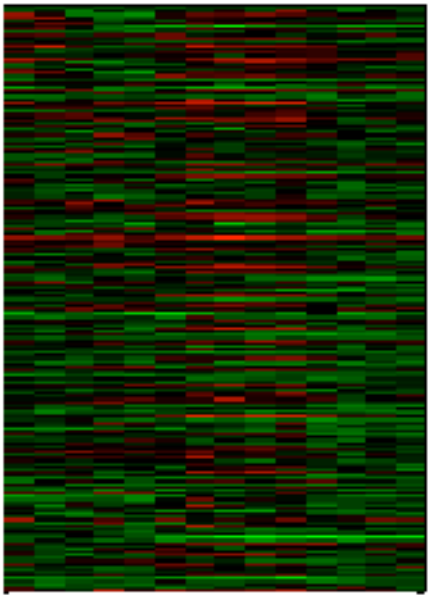


Expression of genes on the left in MD assays

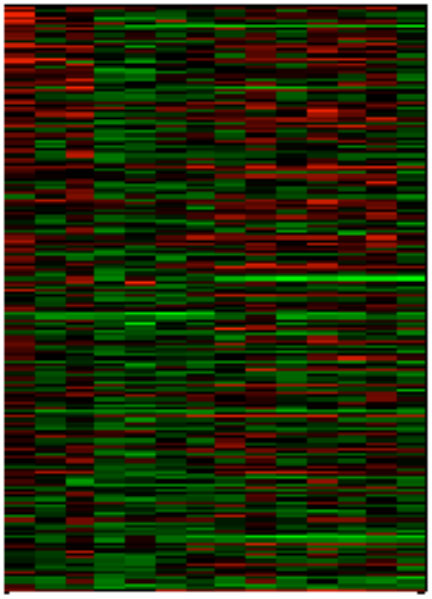


400 Strongest Genes from Genelet 5 Using MD Ordering

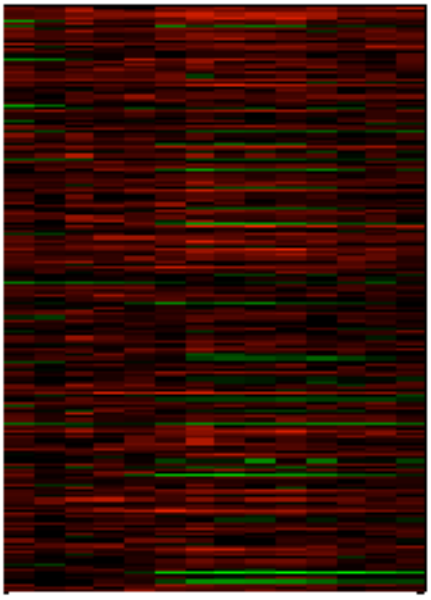
H2O2 Genes parallel to Genelet 6



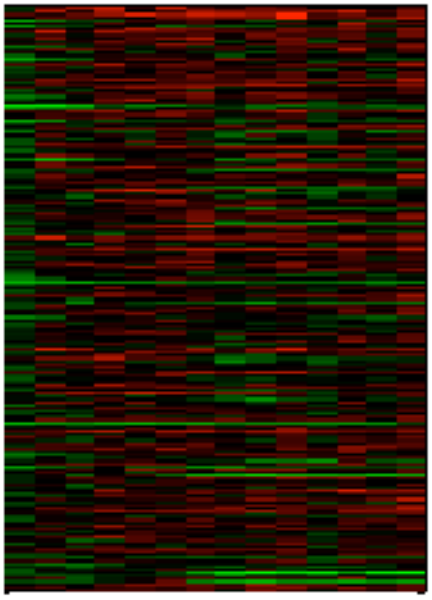
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 6

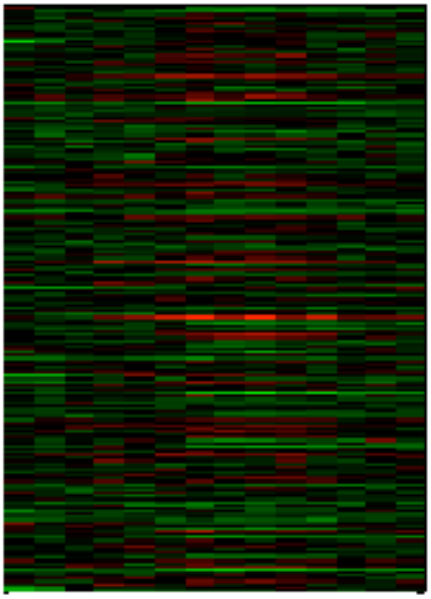


Expression of genes on the left in MD assays

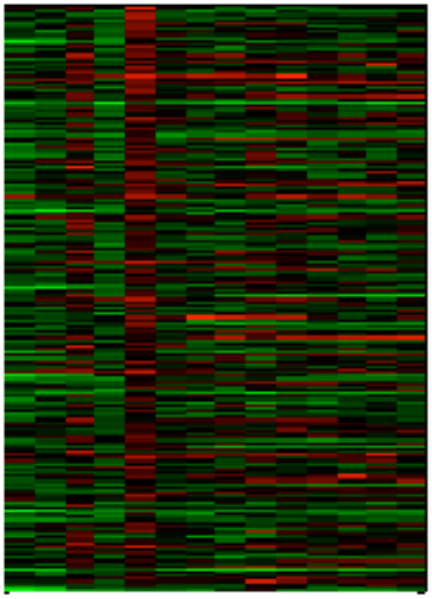


400 Strongest Genes from Genelet 6 Using MD Ordering

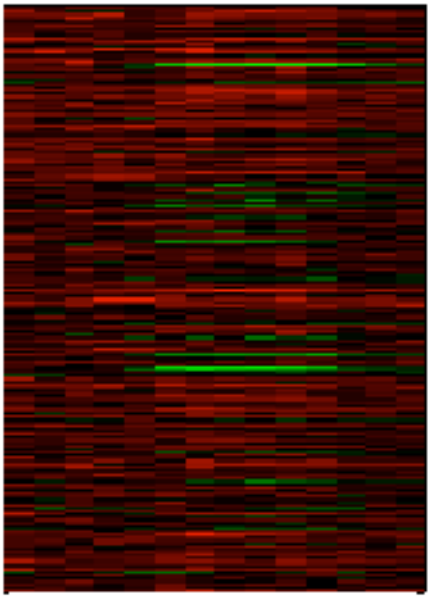
H2O2 Genes parallel to Genelet 7



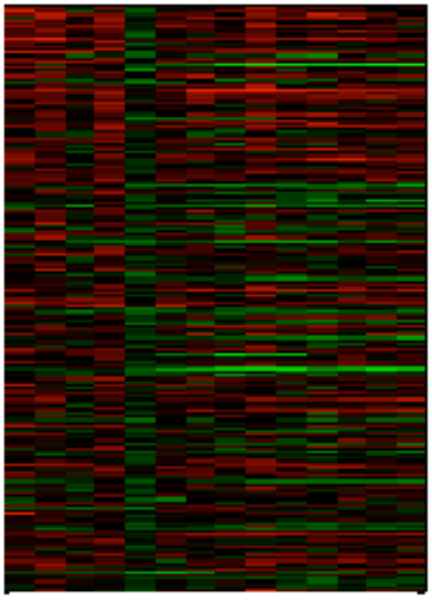
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 7

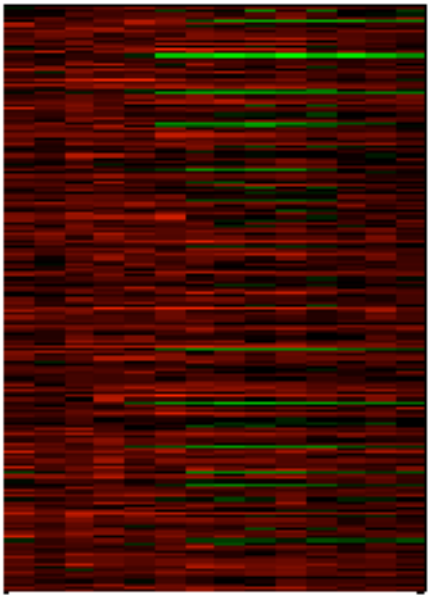


Expression of genes on the left in MD assays

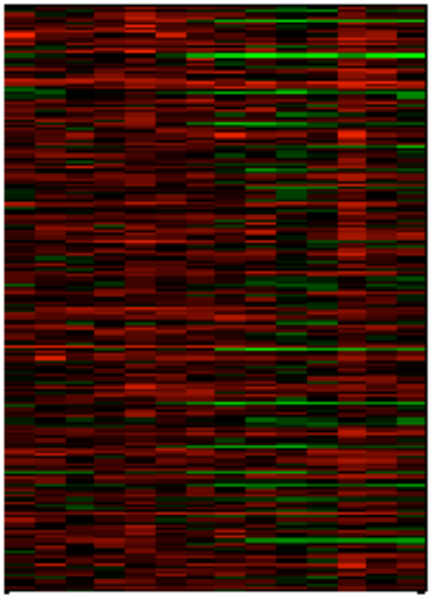


400 Strongest Genes from Genelet 7 Using MD Ordering

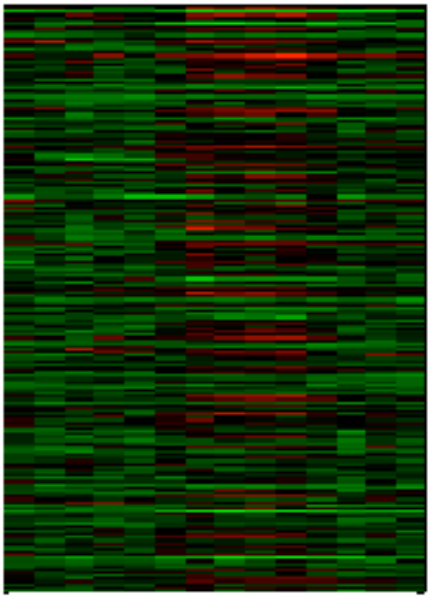
H2O2 Genes parallel to Genelet 8



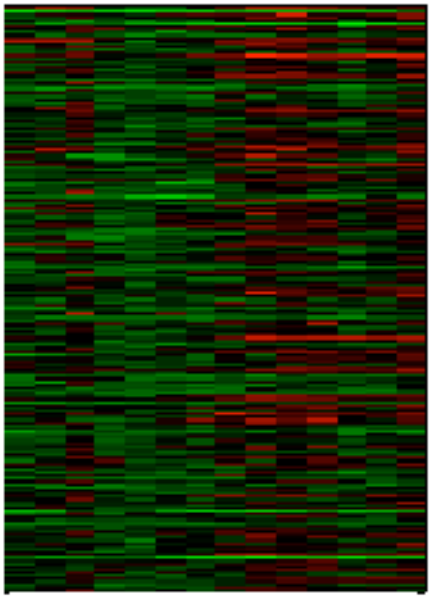
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 8

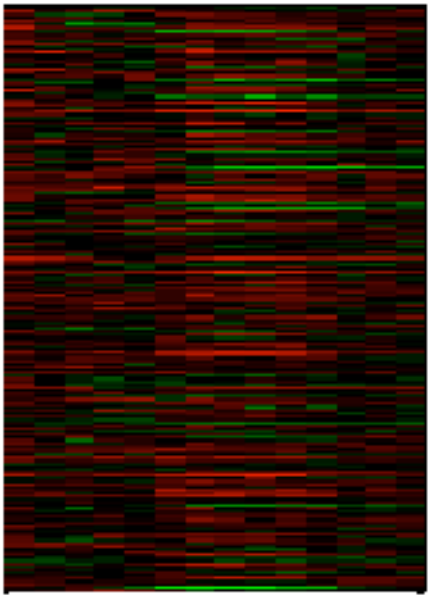


Expression of genes on the left in MD assays

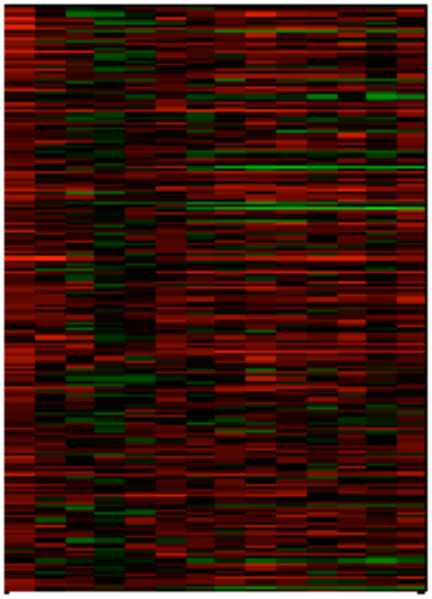


400 Strongest Genes from Genelet 8 Using MD Ordering

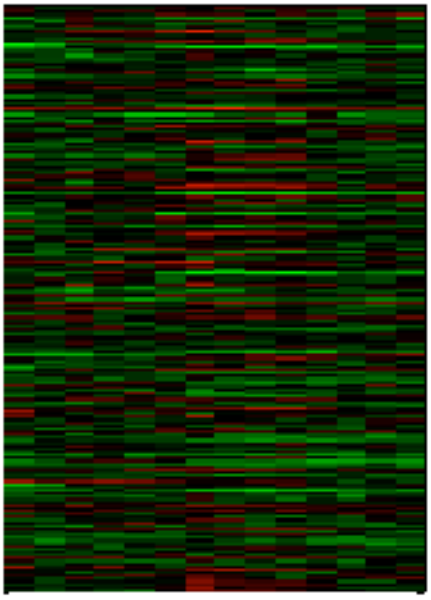
H2O2 Genes parallel to Genelet 9



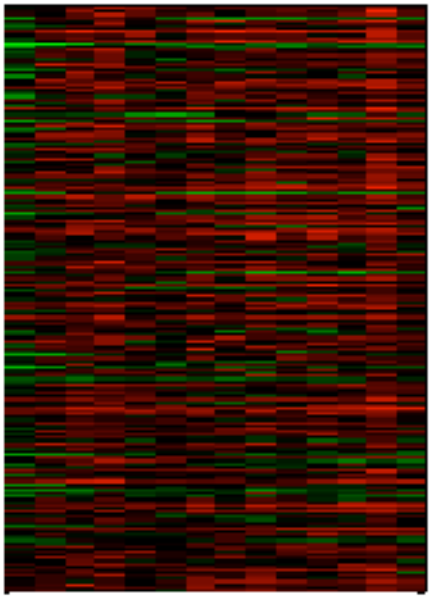
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 9

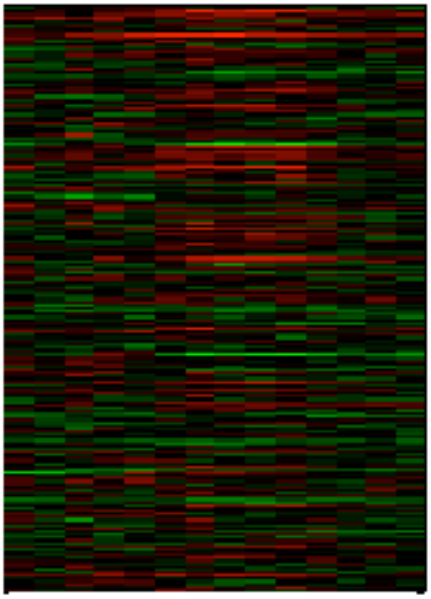


Expression of genes on the left in MD assays

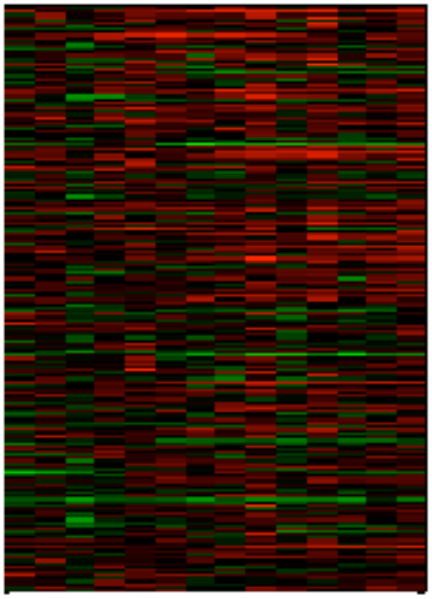


400 Strongest Genes from Genelet 9 Using MD Ordering

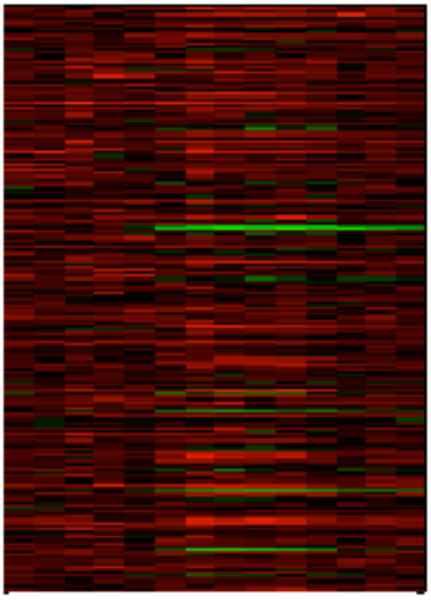
H2O2 Genes parallel to Genelet 10



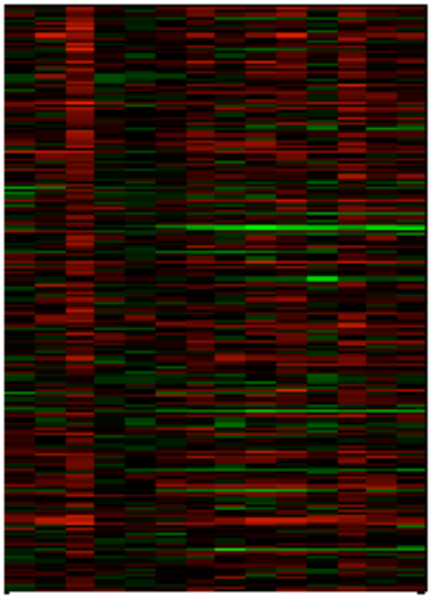
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 10

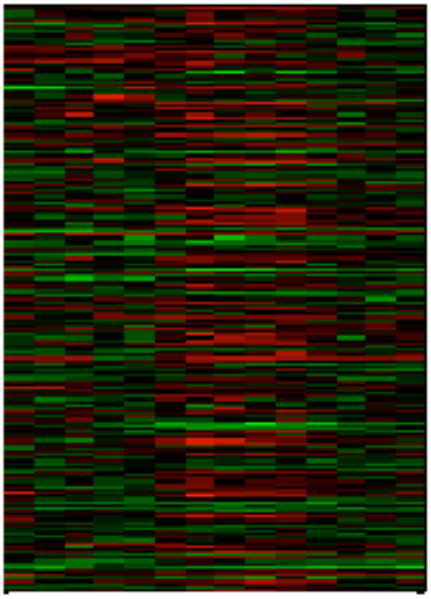


Expression of genes on the left in MD assays

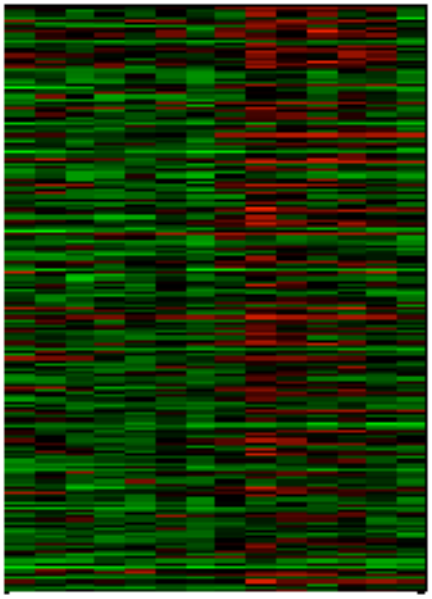


400 Strongest Genes from Genelet 10 Using MD Ordering

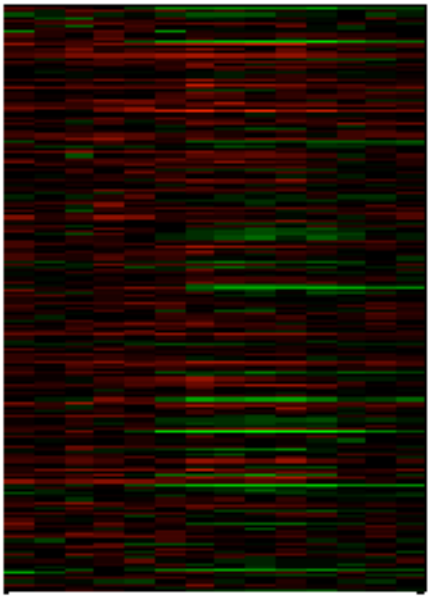
H2O2 Genes parallel to Genelet 11



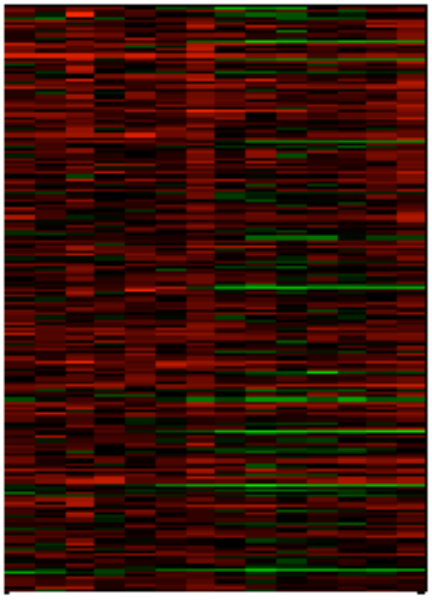
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 11

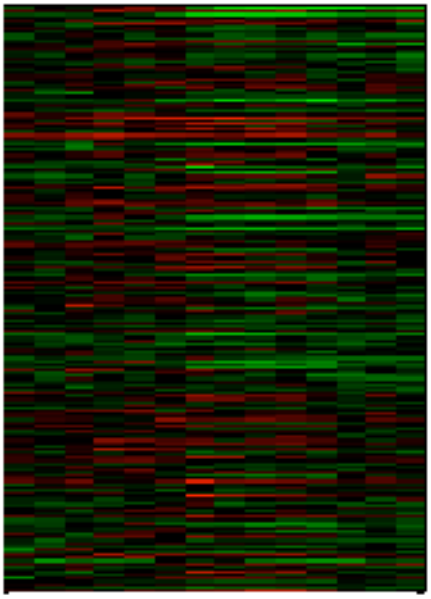


Expression of genes on the left in MD assays

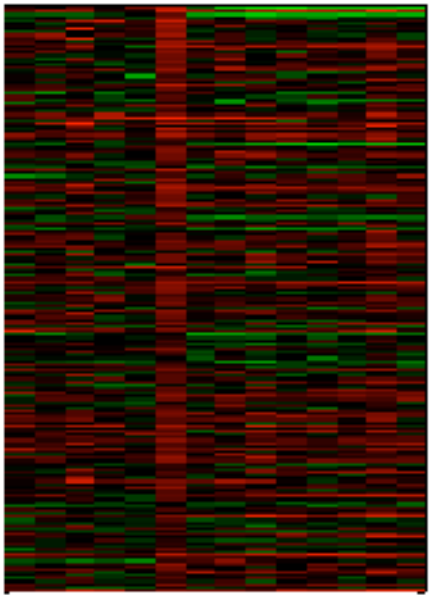


400 Strongest Genes from Genelet 11 Using MD Ordering

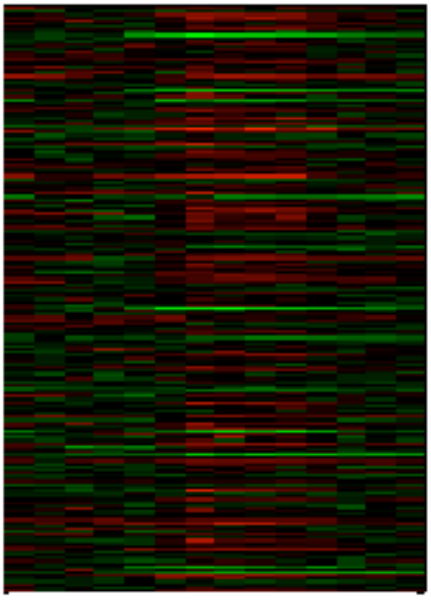
H2O2 Genes parallel to Genelet 12



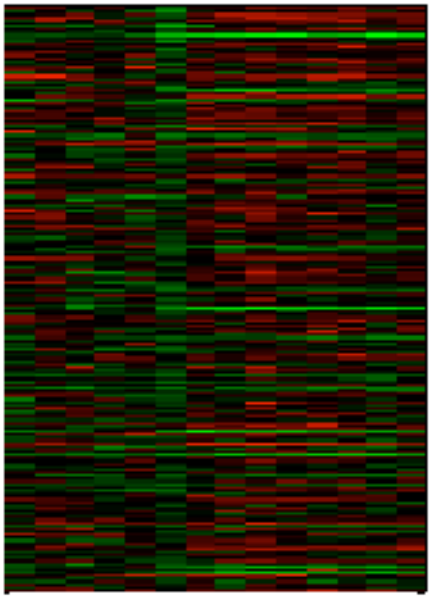
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 12

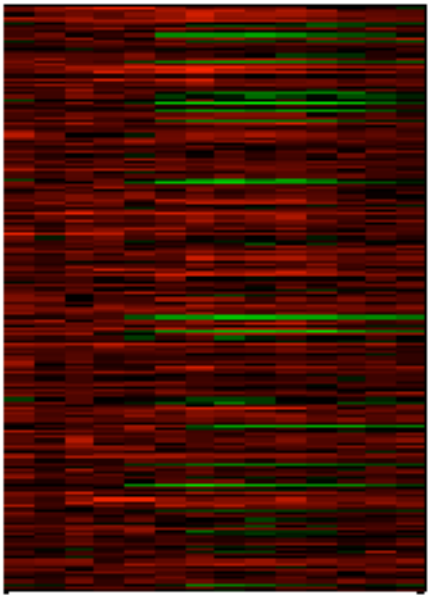


Expression of genes on the left in MD assays

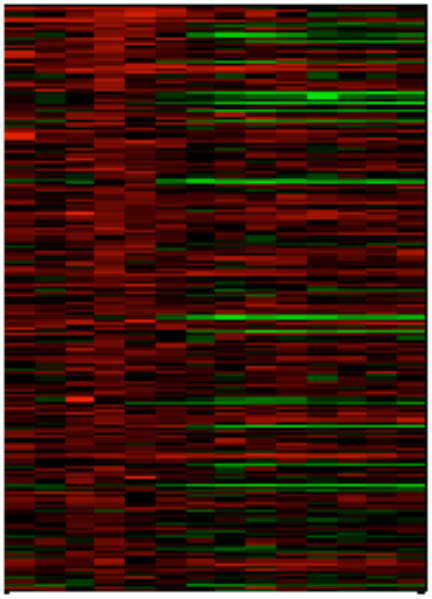


400 Strongest Genes from Genelet 12 Using MD Ordering

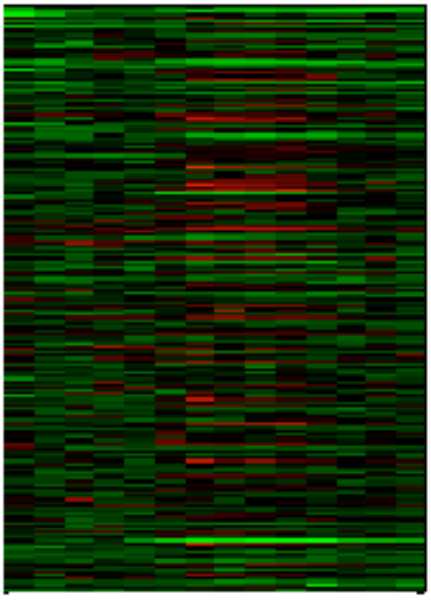
H2O2 Genes parallel to Genelet 13



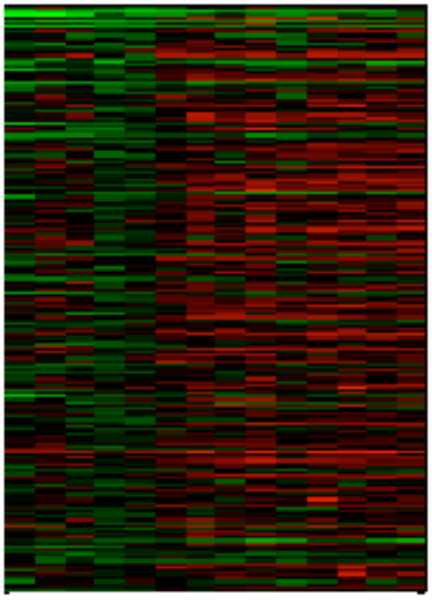
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 13

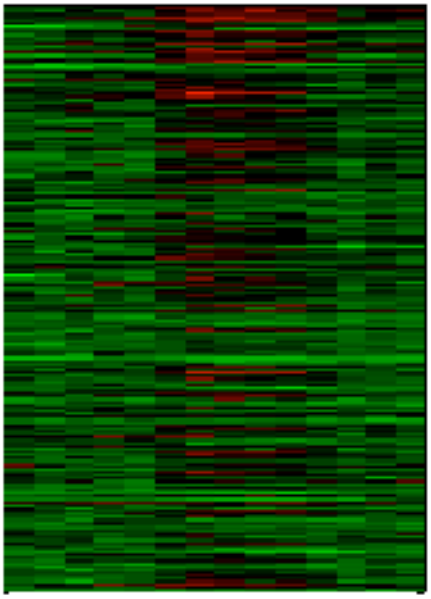


Expression of genes on the left in MD assays

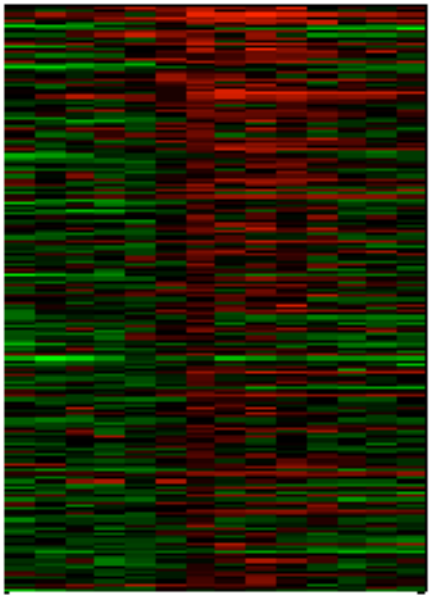


400 Strongest Genes from Genelet 13 Using MD Ordering

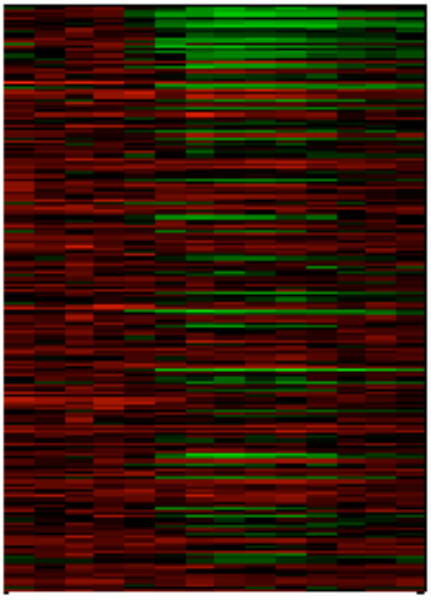
H2O2 Genes parallel to Genelet 14



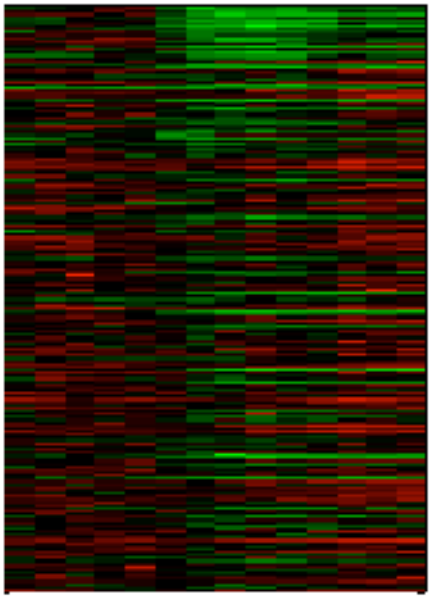
Expression of genes on the left in MD assays



H2O2 Genes antiparallel to Genelet 14



Expression of genes on the left in MD assays



400 Strongest Genes from Genelet 14 Using MD Ordering