

Time-Window Analysis of Developmental Gene Expression Data with Multiple Genetic Backgrounds

[Extended Abstract]

Tamir Tuller ^{*}, Efrat Oron ^{**}, Erez Makavy ^{***}, Daniel A. Chamovitz [†], and Benny Chor [‡]

Tel-Aviv University, Tel-Aviv 69978, Israel.

Abstract. We study gene expression data, derived from developing tissues, under multiple genetic backgrounds (mutations). Motivated by the perceived behavior under these background, our main goals are to explore *time windows questions*:

- 1) Find a large set of genes that have a similar behavior in two different genetic backgrounds, under an appropriate time shift.
- 2) Find a model that approximates the dynamics of a gene network in developing tissues at different continuous time windows.

We first explain the biological significance of these problems, and then explore their computational complexity, which ranges from polynomial to NP-hard. We developed algorithms and heuristics for the different problems, and ran those on synthetic and biological data, with very encouraging results.

1 Introduction

A major goal of systems biology is to infer the relationships among genes and proteins in the cell and organism. A large number of works have tried to identify genes that appear to be coexpressed, in an approach known as "guilt by association". These works come in roughly three major flavors - clustering (*e.g.* [3]), biclustering (*e.g.* [6]), and methods for model inferring (*e.g.* [7, 9]). The problems we deal with in this paper include ingredients from all three. The inputs to our problem include gene expression datasets from two genetic backgrounds. In the first set of problems, the goal is to identify sets of genes with a similar behavior in two equisize subsets of the conditions in the two datasets. Biologically, we are interested in the case where one dataset was generated when a component of the system underwent mutation, while the other dataset represents the wildtype. We focus on developmental gene expression datasets [2], where the conditions are

^{*} School of Computer Science, corresponding author, tamirtul@post.tau.ac.il

^{**} Department of Plant Sciences, oronefra@post.tau.ac.il

^{***} School of Computer Science, erez04@kadant.com

[†] Department of Plant Sciences, dannyc@tauex.tau.ac.il

[‡] School of Computer Science, benny@cs.tau.ac.il

ordered in time, and the subset of conditions of interest are continuous time windows. This natural restriction makes some variants of our problems polynomial. In our experiments, a few key mutations were induced (separately) in a component of a central protein complex. These mutations caused various changes in the behavior of many genes. Some of these changes are best described across contiguous time intervals. This motivates us to define and explore such time interval questions to better understand the functional relations among participating genes. We call these problems “time windows problems”, since we want to find a set of genes and a time window such that the genes’ behavior during this time window in the wildtype is similar to their behavior in the mutant in a different second time window, namely each mutation causes a “time shift” in the expression levels when compared to wildtype. For example, suppose a mutation inhibits the expression of a set of genes, such that it remains 0 in all time points. This phenomenon can cause time shift in the expression level of another gene set. Figure 1 illustrates such a hypothetical example, where a mutation in one gene causes a shift in the expression level of another gene. Gene g is regulated by genes pg_1 and pg_2 according to the table in figure 1A. A mutation causes gene pg_1 to stay at level 0. According to the regulation table, the expression level of gene g at later developmental stages in the mutation (figure 1C) is similar to its expression level in earlier developmental stages in the wildtype (figure 1B). By grouping together genes which exhibit similar shifts in the mutant gene expression compared to the wildtype gene expression, and by combining information about the functionality of some of these genes, one can conclude about the functionality of a mutated gene network (or a mutated protein complex), and the way such a network may regulate directly or indirectly these shifted genes. Thus part of our goals is to find subsets of genes with the same GO annotations [1] and a similar shift.

The second problem of interest is finding a model approximating the dynamics of a gene network in certain continuous time windows. We want to find the regulatory rules of genes by other genes in these time windows. For example, in this work, for the mutant dataset we have the expression levels of several thousand genes at just three time points, while for the wildtype dataset, we have the expression levels of several thousand genes at a few dozen time points. We want to understand the dynamics of the genes in continuous time windows in the wildtype around the time points sampled in the mutant. We employ linear models, and develop heuristic for solving this problem, while being careful to avoid over-fitting.

A number of works have examined analyzing developmental gene expression datasets and comparing gene expression in multiple genetic backgrounds. Arbeitman *et al.* [2] report the gene expression patterns for nearly one-third of all *Drosophila* genes during a complete time course of development. We use this dataset here. Chang *et al.* [5] performed a quantitative inference of dynamic regulatory pathways via microarray data. They used a second order model of differential equations with many parameters, combined with maximum likelihood methods for inferring a regulatory pathways. McDonal and Rosbash [10],

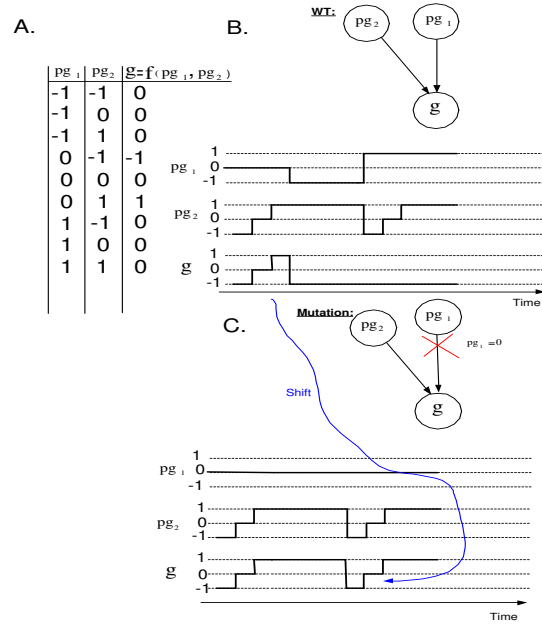


Fig. 1. Illustration of a gene which was shifted by a mutation. A. The expression level of gene g as a function of its regulators, the genes pg_1 and pg_2 , where -1 denote under-expression, 0 denote normal expression, and 1 denote over-expression. B. Hypothetical developmental expression level of gene g in the wildtype. C. Hypothetical developmental expression level of gene g when a mutation causes gene pg_1 to be stuck in level '0'.

developed methods for identification of genes with cyclic behavior while studying circadian rhythms. D'haeseleer and Fuhrman [8] suggested modelling a gene network by a linear model. We note that the main drawback in their approach is the large number of parameters (compared to the size of the dataset) they used, which may lead to over-fitting.

2 Time-Windows Problems: Definition and Mathematical Properties

In this section we present the time shift problems, and deal with their mathematical properties. For lack of space, the proofs are deferred to the full version of this paper. Let S be a set of genes. Let M_1 and M_2 be two gene expression datasets for S , and m_1 and m_2 denote the number of conditions in M_1 and M_2 , respectively. Let $d_S : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^{\geq 0}$, where \mathcal{M} is the space of datasets over S , denote a measure for the *dissimilarity* of the expression pattern of the gene set S in M_1 vs. M_2 . The problems in this work have the following general structure:

Problem 1. Input: Two gene expression datasets, M_1 and M_2 , over a gene set S ; a positive number, δ , and a dissimilarity measure $d_S(M_2, M_1)$.

Task: Find a maximum subset of genes, $S' \subseteq S$, such that $d_{S'}(M_2, M_1) < \delta$.

For a specific example, suppose M_1 and M_2 have the same number of conditions. For every gene g , $M_1(g)$ and $M_2(g)$ are real vectors of the same length. Let $\| \cdot \|_p$ denote the ℓ_p norm. Then $d_{S,p}(M_1, M_2) = \sum_{g \in S} \|M_1(g) - M_2(g)\|_p$ is such a dissimilarity measure. A second example is parameterized by an integer k . For every choice of k conditions C_1, C_2 from M_1 and M_2 , respectively, we look at $M_{1,|C_1}, M_{2,|C_2}$ (the restriction of each dataset to the respective k conditions). Then $d_{S,k,p}(M_1, M_2) = \min_{|C_1|=|C_2|=k} \sum_{g \in S} \|M_{1,|C_1}(g) - M_{2,|C_2}(g)\|_p$ is a dissimilarity measure. In this paper, we deal with problems where the conditions are ordered, usually by time, so that this order has a biological meaning. One example of such order is developmental gene expression dataset, where the i -th condition (column) refers to time t_i , and $i > j \iff t_i > t_j$. We emphasize that t_i in the two dataset need not be the same. In the first problem we use $d_{S,k,2}(M_1, M_2)$ as the dissimilarity measure. We further restrict here C_1 and C_2 to be continuous time windows in M_1 and M_2 , respectively, and $k = m_2$ is the total number of the conditions in the smaller dataset (say M_2). We denote this dissimilarity measure $D_S^1(M_1, M_2)$. For $S = \emptyset$ we define $D_S^1(M_1, M_2) = 0$.

Problem 2. Time shift.

Input: Two ordered gene expression datasets, M_1 and M_2 , where the number of conditions in M_1 is larger than in M_2 ($m_1 \geq m_2 = k$), and a positive number, δ .

Task: Find a maximum set of genes S' and a continuous time window in M_1 , $W_1 = i_1, \dots, i_k$, such that the expression of the gene in this window is similar (error less than δ) to their expression level in all the conditions of M_2 . Quantitatively $D_{S'}^1(M_2, M_1) < \delta$.

A generalization of problem 2 with allows k (the size of the windows) to be smaller than both m_1 and m_2 . The windows should still be continuous. In this case, we denote the dissimilarity measure $D_{S,k}^2(M_2, M_1)$. In the second problem we want to infer models, which describe the behavior of the genes in M_1 (the larger dataset). We allow different models for different time windows in M_1 , while we focus on "interesting" time windows, that contain the time points in M_2 (each time window in M_1 is around different time point in M_2). The simplest such model is a linear model.

D'haeseleer and Fuhrman [8] were the first to suggest the use of linear models for analyzing gene networks. Regulation of genes can be described as a function of the expression levels of other genes by a differential equations. These equations can be approximated by difference equations which can be described by an equivalent set of linear equations. More generally, the behavior of many dynamic models at time $t + \Delta t$ can be approximated by a linear function of the model's parameters at time t . The relative error is the ratio between the error and the average gene expression level. In our model we got better results in terms of relative error when we worked with the logarithm of the expression value. Taking

logarithms when working with gene expression level is justified for example in [12]. So we aimed to express the log of the expression level of a gene at time point i by a linear combination of the log expression levels of a subset of the genes at time $i - 1$ (the “parents” of the gene), such that the expression level of gene g_n at time i equals approximately: $g_n(t) \approx \sum_{g_j \in pa(g_n)} w_{n,j} \cdot g_j(t-1)$, where $pa(g_n)$ denotes the set of “parents” of gene g_n , and $w_{n,j}$ are constants. For each gene we need to find a *different* set of parents and weights, $w_{n,j}$ but these are fixed for all the conditions. The error of a gene according to such model is the Euclidean distance between the predicted vector and the actual vector across the window, W : Let $\hat{g}_n(t) = \sum_{g_j \in pa(g_n)} w_{n,j} \cdot g_j(t-1)$ be the “predicted” value, then the error for gene g_n in window W is: $e_{g_n, W} = \sqrt{\sum_{i \in W} (g_n(t) - \hat{g}_n(t))^2}$. This roughly describes the model of [8]. We deal here with a more general model: First, we want to find different linear models for different continuous time windows of development, in that we are looking for a phase, or window, dependent network. We define the error of a set of genes in a time window to be the error (e_{g_n}) of the gene with the largest error in this set in the time window. As defined, this problem has too many degrees of freedom in choosing the parents’ sets of each genes, resulting in models that are often meaningless [13]. Thus we want to bound the maximum in-degree in the linear network. The bound should be smaller than the number of conditions in that window.

Thus by [13] if we want to describe an ℓ dimensional vector by a linear model where its parent set are of ℓ or more vectors in the ℓ -dimensional space we can get zero error by using random vectors with probability approaching 1 as $k \rightarrow \infty$. If the number of parents is smaller than ℓ , this phenomenon does not occur. Thus if our vector describes a gene expressions, such model may reveal a true relation between a gene and its parents. This motivates us to restrict the number of parents of each gene in the model to be smaller than the number of time points in the window. By using this upper bound, we prevent over-fitting a model to a window. Thus, we are interested in the following problem:

Problem 3. Linear approximation with bounded in-degree.

Input: Two ordered gene expression datasets M_1 and M_2 , a positive number, ε , two positive integer h and W .

Task: For every point in M_2 find a linear model with in-degree less than h for the set of genes in M_1 , and for the window of size W around the point. When the expression level of the set in M_1 is described by this model, its error is required to be smaller than ε for the window.

The restriction of bounded in degree makes the problem computationally hard. In the decision version of problem 3 we have the same inputs, and have to decide if there is a model with in degree less than h that approximate the genes in M_1 in windows around the points of M_2 with error less than ε .

Lemma 1. *The decision version of problem 3 is NP-hard.*

For a given h (if h is not an input of the problem) problem 3 is practical only for $h \leq 3$ (by exhaustive search in complexity $O(n^h)$).

The following lemma indicates that if our dataset was sampled from a linear model, when using this dataset for inferring a linear model in a time window where many genes change slowly, we should expect that the parents' set of a gene will contain genes which are not directly connected to the gene in the real model (but are close to it in the real model). It is known that other model inferring methods suffer from similar problems. However, when our method missed edges of a gene's "real parents", it often replaces them by edges from the "grandparents generation". This phenomenon is tolerable since we get sets of genes that are relatively close to a gene's parental set. Let $gp(g_n)$ denote the set of grandparents of the gene g_n in a linear model. We say that a gene g_k changes slowly if $(1 - \varepsilon) \cdot g_k(t - 1) \leq g_k(t) \leq (1 + \varepsilon) \cdot g_k(t - 1)$, where $0 < \varepsilon \ll 1$. The following lemma explains this phenomena. By recursively using the arguments of lemma 2 we can get similar results for the connection between the expression level of a gene and its ancestral of depth d . In this case we will get the approximating factors $(1 - \varepsilon)^d$ and $(1 + \varepsilon)^d$ instead of $(1 - \varepsilon)$ and $(1 + \varepsilon)$, which increase the error exponentially (with d). It easy to see that when the average in-degree in the net is larger than 1, replacing the real parents of a gene by its ancestors implies an increase of its in-degree.

Lemma 2. *For the linear model with bounded degree, if the gene's grandparents change slowly, the optimal set of parents for a gene can be well approximated by the set of its grandparents.*

3 Algorithms and Heuristics

The time shift problems have two stages: Finding a set of shifted genes, and identifying a subset with functional enrichment (GO annotation) in each such a set. Let $M_{r,|C_j,i}(g)$ denote the i -th sample of gene g in time window C_j of dataset M_r . A direct calculation of the cost function for the two variants of the time shift problems for a gene g , when comparing the time windows C_2 and C_1 (of size k) in M_1 and M_2 respectively, is:

$$\|M_{2,|C_2}(g) - M_{1,|C_1}(g)\|_2 = \sqrt{\sum_{i=1}^k (M_{2,|C_2,i}(g) - M_{1,|C_1,i}(g))^2}.$$

Subtraction and addition are much cheaper processor operations, compared to squaring and square root operations. This calculation involve performing k squaring and one square root operations, namely $k + 1$ "expensive" operations. In the naive way of finding solutions to the time shift problem, we separately calculate for each gene the cost function using the above equation, and attribute it to the pair of windows C_1, C_2 if this function is less than δ . In other words, for each gene we need to calculate the cost function $(m_1 - k - 1) \cdot (m_2 - k - 1)$ times, for each pair of time windows $C_1 \subseteq M_1$ and $C_2 \subseteq M_2$. In total we have $(m_1 - k - 1) \cdot (m_2 - k - 1) \cdot (k + 1)$ expensive operations for just one gene.

Let $M_{r,i}(g)$ denote the expression level of gene g in dataset r in time point i . In order to speed up the process, we do the following: For each gene, we first calculate a table of size $m_1 \cdot m_2$, where the (i, j) entry in the table contains the value $(M_{2,i}(g) - M_{1,j}(g))^2$. We use the fact that each entry in the table is used

for many pairs of windows, and use the values in the table for calculating the cost function for different pair of windows for g . Since calculating the table costs us $m_1 \cdot m_2$ expensive operations and by using the table we only need to perform one expensive operation (square root operation) for the calculation of the cost for a pair of windows, we now perform a total of $m_1 \cdot m_2 + (m_1 - k - 1) \cdot (m_2 - k - 1)$ expensive operations. Asymptotically (for large m_1 and m_2) this is $\Theta(k)$ faster.

In the next stage we searched for functional enrichment in the solution set, to better understand possible biological meanings of our results. We used GO annotations [1] which attribute genes to cellular functions. We are interested in sets of genes that have both a similar time shift and a common function, as determined by GO annotation. Let G denote a bound for the maximum number of GO annotations for a gene. We calculated the number of genes with each GO annotation in our dataset by one pass over all the genes, and for each gene we checked at most G annotations (a gene may have more than one annotation, and we assume no more than G). We generated a table with the number of genes with each GO annotations. The overall complexity of this stage $n \cdot G$. Given a set of size $|S|$ of shifted genes, we generated a similar "small" table only for the set, in time complexity $G \cdot |S|$. These tables enable us to calculate the enrichment's p-values, using the standard formula of hypergeometric distribution.

We now turn to the linear model problem. Since the bounded linear model problem is NP-hard, we used variations of the following heuristic. Let m_1 denote the number of time samples in M_1 . Check all the $m_1 - k - 1$ continuous windows of size k , the specified window size. For each such window, perform the following greedy heuristic for each gene g_n :

1. Start with an empty set of "parents" for g_n .
2. At step r , by exhaustive search, find the gene that causes the largest decrease in the prediction error of gene g_n when adding it to the $r - 1$ current parents of the gene g_n , add it to the parents set of the gene.
3. Stop if the decrease is less than $\alpha \cdot \hat{\varepsilon}_{r,n}$ or the number of parent is larger than m_1 .

The number α is a parameter to our algorithm, and $\varepsilon_{r,n}$ denotes the average decrease in the error of gene g_n with $r - 1$ parent that were found by the greedy algorithm, when adding a random r -th parents to the gene. Let $\hat{\varepsilon}_{r,n}$ denote an estimation of $\varepsilon_{r,n}$. For each r ($1 \leq r < k$) and gene, g_n , in the dataset M_1 , $\varepsilon_{r,n}$ is estimated empirically during the above exhaustive search, by averaging the decrease in the error in r -th step. *I. e.* when adding a gene to the parent set of size $r - 1$ of g_n . In multiple regression we fit the coefficient $w_{n,j}$, of the gene's "parents" in the linear model. This is done by finding $w_{n,j}$, which minimize the error e_{g_n} (by differentiating and comparing it to zero). In stage 2 of the algorithm, we perform multiple regression for each candidate set of parents. Let k (the size of the window) denote an upper bound on the number of parents for each gene in the model. Let $C_{mk}(k)$ denote the time complexity for calculating multiple regression with k variables. In our case this equals the complexity of inverting a $k \times k$ matrix, which is $O(k^3)$ practically. Let n denote the number

of genes in the dataset. The overall time complexity of the algorithm for a given time window is $O(k \cdot C_{mk}(k) \cdot n^2) = O(k^4 \cdot n^2)$. As in the previous problems, here we also used the fact that models of close windows are similar. For each gene we kept the parents set which our algorithm found for the closest previous window and tried to find a better one. To avoid local maxima, for each gene we tried to optimize its parents set by checking different subsets from the set of genes which give large decrease in the error in the initial stages of the algorithm.

4 Results

Our data consist of developmental gene expression datasets of the fruit fly *Drosophila melanogaster*. One was wildtype dataset and others were from different mutants in the Cop9 signalosome. The Cop9 signalosome (CSN) is a highly conserved protein complex, conserved across different organisms and known to be essential for development of plants and animals. CSN has eight subunits that regulate multiple signal transduction pathways [4]. These subunits are inter-related, and some are found in multiple configurations [11]. Consequently, the biological roles of the complex as a whole, and of individual subunits, are not completely understood. To clarify this situation, we are employing transcriptional profiling on *Drosophila csn* mutants. Four mutants in different CSN subunits were analyzed at three developmental time points: 60, 72, and 96 hours after egg deposit (AED). This is the first global comparison between multiple CSN mutants in animals, and as such we expect it to shed light on CSN involvement in unknown processes, and lead to new and improved models for the role of CSN and its subunits. We analyzed our data together with publicly available wildtype samples of Arbeitman *et. al* [2], containing 80 time points.

4.1 Detecting putative time-shifted and partially time-shifted genes

Some of the results for putative time-shifted genes compared to all the three time points in the mutants (problem 2) are summarized in table 4.1. In table 4.1 a gene is attributed to a shift bin $[A, B]$ if it there is at least one $A \leq \Delta \leq B$ such the gene exhibits at time t in the mutant expression pattern as in time $t - \Delta$ in the wildtype. For the four mutants analyzed, 120 sets of genes with suspected time shifts were identified. In a global look at the data, we first notice that while in mutants 3 and 4 the number of negative and positive shifts was equal (14 : 16 and 16 : 14 respectively), in mutant 2, and especially mutant 1, most of the shifts were negative (21 : 9 and 25 : 4, respectively). This may suggest that mutations 1 and 2 caused late-acting genes to be induced earlier. Table 4.1 shows a sample of these sets with their predicted shifts and accompanying P-value. The functional analysis of these genes indicates that most of these sets are involved in various aspects of development and cellular regulation, such as regulation of DNA structure and integrity (rows 3, 5, 6, 8) and signal transduction (rows 1, 2, 9, 12, 14). Only a few sets are obviously involved in "house keeping" functions. For example, three genes, whose gene products are

all involved in glycolysis, were found to have a bin shift of $[-4, 4]$ (line 4). These genes comprise 1/3 of the genes with a similar bin shift, but only 0.49% of total genes used in the analysis. The $[-4, 4]$ bin basically represents genes for whom no significant shift is found. Glycolysis, the breakdown of glucose to usable energy forms, is the one metabolic pathway that occurs in all living cells and is the starting point for aerobic respiration and fermentation. As such, no effect of the CSN on glycolysis was expected *a priori*, and indeed, this is illustrated in this example. Interestingly, mutants 3 and 4 both show bin shifts with genes encoding subunits of the proteasome (lines 11, 15, 17), and the proteasome regulatory lid in particular (lines 11 and 17). The proteasome is a large multiprotein complex that degrades proteins in regulated fashion. That these genes are regulated by the CSN is interesting as the proteasome lid is evolutionarily related to CSN, and the lid and CSN interact physically to regulate similar processes. As yet, there is no in depth understanding on the cross-talk between these two complexes, nor is the regulation on the lid clearly understood. However, finding that the regulated expression of these genes is shifted in a COP9 signalosome mutant provides further evidence for the mutual dependence of these complexes.

In the next stage we deal with windows of size 1. We observed that in all mutants at time 60 AED, more genes were up- than down regulated in relation to the wild type. This expression pattern may be explained by the corresponding mutations causing loss of function of transcriptional repressors. We hypothesized that these genes are either up regulated in the mutant before they would normally be so in the wildtype (that is in the wildtype they should be up regulated at $t = 60+$ shift), or alternatively that these genes are normally upregulated early in development, but then not repressed in the mutant (that is in the wild type they should be upregulated at $t = 60-$ shift). Our results for putative time-shifted genes compared to only one time point 60 in the mutants (problem 2, where the conditions of the mutants include only time point 60) are summarized in table 2. Table 2 shows that both types of behaviors were identified. The first two rows of Table 2 show genes that are up regulated in two CSN mutants at 60 hours, while in wildtype these genes peak during early or late embryogenesis (0 – 24 hrs AED). At the other extreme, the last two rows in Table 2 show sets of genes that are induced in a CSN mutant at 60 hours, while in the wildtype, these genes normally peak either during metamorphosis (149 – 161 hrs AED) or in old adults (527 – 827 hours from hatching).

4.2 Intermittent linear model: Synthetical and Biological inputs

To evaluate our method, we first checked our algorithm on small nets (containing a few dozen genes and conditions). We sampled known nets and tried to reconstruct them by our and by D’haeseleer’s algorithms. We counted the number of real edges each algorithm missed, and the number of edges that do not exist in the real model and each algorithm adds. The simple least square fit method was substantially worse than our method, it missed 42% more real edges than our method, and it add 425% more false edges compared to our method. We then ran our procedure on real dataset of Arbeitman *et al.* [2] with 4000

genes, and generated linear models for three time windows as output. The first window was for the times window 24 – 105, the second for times 19 – 57, and the third was for times 67 – 113 (around the time points in the datasets of our mutants). Each window contained ten samples. For lack of space, we describe here the results for only one sub graph constructed model. Further analysis is deferred to the full version of this paper. Figure 2 describes the sub graph of the constructed models for a small set of chosen genes. Figure 2 shows an analysis of a small gene network, where gene 1 encodes a transcription factor known to be involved in a developmental process, gene 4 encodes a hormone receptor involved in this process, and genes 2, 3, 5, and 7 are known to be regulated in this process, though their connection to 1 and 4 is unknown. Our linear modeling correctly identifies gene 1 as a key node in this network, where it regulates the other members of this network, with the exception of gene 7, which appears as an "orphan". Interestingly, early in development we identify a putative feedback inhibition loop between the transcription factor (1) and the activating receptor (4). In late development, gene 2, whose biochemical and developmental function are unknown, has an inhibitory effect on the network, negatively affecting both the transcription factor (1) and another gene (3).

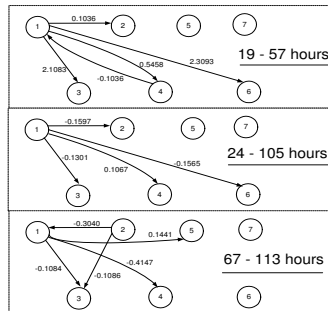


Fig. 2. Description of the dynamics of a gene set at three time windows around time 60 - 96. The figure describes only the sub-model for this set, edges from/to other genes were omitted.

5 Conclusions and Further Research

In this work we investigated problems originating from developmental gene expression datasets of multiple genetic backgrounds. We defined two major questions, explained their biological significance, and their mathematical properties. One of the problems is polynomial, while the other is NP-hard. We developed algorithms for solving two variants of first one, and a heuristic for the other. We implemented and ran them on synthetic and biological inputs. Our methods

generated many interesting biological results, some exhibiting agreement with the acceptable biological knowledge. This supports the underlying reasoning of our approach. There are many open questions and directions we are considering. Here we describe two of them. First, more biological experiments are underway, in order to achieve a richer dataset for our mutants (a dataset with more time points). Such datasets will enable us to infer linear models for the mutants and compare these models to the one inferred for the wildtype. This way, we could explore a new time window problem, where we compare models from different genetic backgrounds in different time windows. Another direction involves inferring linear models from datasets of multiple species, an approach that may help filtering noise and avoid over-fitting.

Acknowledgements

We wish to thank Drs. Bruce Edgar and Ling Li of the Fred Hutchinson Cancer Research Center for providing the facilities for and assistance in carrying out the microarray hybridizations and Dr. Daniel Yekutieli from Tel Aviv University for helpful discussions. This work was partially supported by grants from the Manna Institute (EO) and Israel Science Foundation (DAC, BC).

References

1. GO annotation guide. <http://www.geneontology.org/go.annotation>.
2. N. M. Arbeitman, M. E. E. Furlong, F. Imam, E. Johnson, H. B. Null, S. B. Baker, A. M. Krasnow, P. M. Scott, W. R. Davis, and P. K. White. Gene expression during the life cycle of drosophila melanogaster. *Science*, 297:2270–2275, 2002.
3. A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6:281–297, 1999.
4. DA. Chamovitz and A. Yahalom. A systems approach to the cop9 signalosome. *Plant Physiol*, 132:426–427, 2003.
5. W. C. Chang, C. W. Li, and B. S. Chen. Quantitative inference of dynamic regulatory pathways via microarray data. *BMC Bioinformatics*, 6(44), 2005.
6. Y. Cheng and G. M. Church. Biclustering of expression data. *In Proc. ISMB'00*, pages 93–103, 2000.
7. B. Chor and T. Tuller. Adding hidden node to gene network. *WABI2004*, 2004.
8. P. D'haeseleer and S. Fuhrman. Gene network inference using a linear, additive regulation model. *Bioinformatics*, 2000.
9. N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian network to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.
10. M. J. McDonald and M. Rosbash. Microarray analysis and organization of circadian gene expression in drosophila. *Cell*, 107(5):567–578, 2001.
11. A. Orian, B. Van Steensel, J. Delrow, H. J. Bussemaker, L. Li, T. Sawado E. Williams, L. W. Loo, S. M. Cowley, C. Yost, S. Pierce, B. A. Edgar, S. M. Parkhurst, and R. N. Eisenman. Genomic binding by the drosophila myc, max, mad/mnt transcription factor network. *Genes Dev*, 17:1101–14, 2003.
12. D. M. Rocke and B. Durbin. Approximate variance-stabilizing transformations for gene-expression microarray data. *Bioinformatics*, 19(8):966–972, 2003.
13. T. Tao and V. Vu. On the singularity probability of random bernoulli matrices. *submitted*, 2005.

No	mutant	GO-ID	Bin Shift Range	Shift Bin Size	Func Bin Size	Genes with this Func	P-value
1	1	7274	[-16 -8]	29	2	3	$2.933 \cdot 10^{-4}$
2	1	19221	[-68 -64]	14	2	11	0.0012
3	1	6333	[20 28]	23	2	13	0.0044
4	1	6096	[-4 -4]	33	3	14	$4.499 \cdot 10^{-4}$
5	2	6398	[-12 -8]	26	2	6	0.0011
6	2	5730	[-104 -100]	40	3	10	$2.73 \cdot 10^{-4}$
7	2	3779	[52 56]	7	2	49	0.0056
8	2	6281	[-64 -48]	16	2	25	0.0078
9	3	8523	[8 16]	30	2	2	$1.057 \cdot 10^{-4}$
10	3	15144	[-96 -92]	28	2	5	$8.94 \cdot 10^{-4}$
11	3	5838	[52 60]	17	2	9	0.0011
12	3	5099	[-104 -100]	82	2	2	$8.07 \cdot 10^{-4}$
13	4	4559	[52 56]	27	2	2	$8.53 \cdot 10^{-5}$
14	4	8195	[24 32]	56	2	2	$3.74 \cdot 10^{-4}$
15	4	8540	[12 20]	48	3	5	$4.26 \cdot 10^{-5}$
16	4	9993	[-16 -12]	14	2	15	0.0022
17	4	5838	[12 16]	53	4	9	$1.2 \cdot 10^{-5}$

Table 1. Representative results from time-shift analysis where the threshold = 0.1. Bin Shift Range shows range of the shift in hours - the difference between the original time point and the shifted time point, for example if the shift is -100 the shifted time point are 100 hours after the original ones, i.e. the mutation caused the gene expression of time $T + 100$ hours to be expressed in time T . Shift Bin Size: Number of genes in this shift. Func Bin Size: Number of genes with this GO-ID in this shift. Genes with this GO-ID: Total no of genes with this GO-ID on the chip, out of 2869 genes with GO-ID. P-value.

No	Bin Shift Range	GO-ID	Shift Bin Size	Func Bin Size	Genes with Func	P-value
1	[8 9]	6139	60	5	7/242	0.0102
2	[19 20]	4702	32	5	9/242	0.0024
3	[149 161]	4674	37	4	6/242	0.0053
2	[527 827]	8248	50	3	3/242	0.0084

Table 2. Representative results from partial time-shift analysis. Bin Shift Range shows the wildtype expression induction range (in hours) for the genes upregulated at 60 hrs AED in the mutants. Shift Bin Size: Number of genes in this shift. Func Bin Size: Number of genes with this GO-ID in this shift. Genes with this GO-ID: Total no of genes with this GO-ID among the 242 genes up-regulated at 60 hrs AED that have a GO-ID and are also present in the wildtype data set.