Exploring Gene Regulatory Networks for Drug Target Identification

Lei Zhang
Siemens Corporate Research
755 College Road East
Princeton, 08540
Lei.Zhang@scr.siemens.com

Bernd Wachmann
Siemens Corporate Research
755 College Road East
Princeton, 08540
Bernd.Wachmann@siemens.com

Julien Etienne
Siemens Corporate Research
755 College Road East
Princeton, 08540
Julien.Etienne@scr.siemens.com

ABSTRACT

Motivation: Gene expression analysis has been widely studied to identify networks and relationships among the genes. The target identification problem is a critical step in drug development to minimize the undesirable side effects of a candidate drug. Clustering and Bayesian methods[5] were applied for the reverse engineering of gene networks; however, it is still challenging to identify the target of a compound (i.e. a protein) among the potentially large amount of genes which change their expression values due to the compound. A major challenge in drug target identification is that, for many drug candidates, the targets are unknown and difficult to distinguish from the thousands of additional gene products that respond indirectly to changes from the activity of the target.

Results: In this paper, we present a computational framework to explore gene regulatory networks for drug target identification. We perform experiments on a publicly available data set containing profiles of gene deletions, titratable promoter insertions and drug compound treatments. We demonstrate our approach achieved accurate results in finding the known targets and associated pathways while being computational efficient.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics

Keywords

Drug Target Identification, Gene Regulatory Network

1. INTRODUCTION

Gene expression analysis has been widely studied to identify networks and relationships among genes. The target identification problem is a critical step in drug development to minimize the undesirable side effects of a candidate drug. A major challenge in drug target identification is that, for many drug candidates, the targets are unknown and difficult to distinguish from the thousands of additional gene products that respond indirectly to changes from the activity of the target[2, 3, 4, 7, 8, 9, 11, 13, 14].

DNA microarray technology provides an opportunity to efficiently identify a compound's target by observing all genes with a transcriptional response to the compound treatment. However, as stated in [4], whole-genome expression profiles do not distinguish the genes targeted by a compound from the indirectly regulated genes. In the past few years, many approaches have been proposed to overcome this problem. The association analysis techniques[9][12][15], haploinsufficiency profiling[7][8][11] and chemical-genetic interaction mapping techniques[14] have been developed, however, they require libraries of genetic mutants or fitness-based assays of drug response. The identification of the complex regulatory networks would be extremely valuable. Clustering and Bayesian methods[5] were applied for the reverse engineering of gene networks; however, it is still challenging to identify the target of a compound (i.e. a protein) among the potentially large amount of genes which change their expression values due to the compound. Model estimation techniques proposed in [6][16] suffered from the requirement of knowledge of the gene targets of each training perturbation. A mode-of-action by network identification (MNI) has been proposed in [4]. Assuming that training profiles are obtained in steady state following a variety of treatment, including compounds, RNAi, and gene-specific mutations, this method was able to use varied treatment types in the training data and improved flexibility thus facilitating application of model based approaches to higher model organisms where gene-specific perturbations are more difficult to implement. An Expectation Maximization (EM) type of iterative procedure was employed in [4] to infer a network model without requiring gene-specific perturbations. However, such an iterative procedure is computational expensive. In this paper, we propose a first order computational framework to infer a network model for drug target identification which is accurate and efficient. Another difference between our method and [4] is that, during the dimensionality reduction step, instead of applying singular value decomposition(SVD) to the whole training set, we pick a small set of highly correlated genes for each gene to learn the network model since we believe that each gene is connected to a small set of other genes in the regulatory network. Since during network modeling we aim to have training experiments with no or small external influences on the network, it is not accurate to ap-

SIGKDD '2006 Philadelphia, Pennsylvania USA

ply global dimensionality reduction techniques. More details and the statistical validation will be provided in Sec. 2.2.

In our method, given the observations of genes with transcriptional responses provided by DNA microarray data, in order to measure the effects of the test compound, we first build a first order network model for each gene using linear regression. Once the regulatory model is trained, the impact of the compound on the expression of a particular gene can be quantified by the ability of the model to correctly predict its expression value and to quantify occurring deviations. The genes are then ranked where the highest and lowest-ranked genes are those whose expression is most inconsistent with the model due to the external influence of the compound on those genes. Compared with [4], instead of using the experimental iterative approach which requires certain convergence criteria, we used a simple two layer network modeling framework, which is efficient. We also measure the difference in both positive and negative directions since the sign of the error is also informative. The experimental results demonstrate that our approach achieves accurate results in finding the known targets and associated pathways while being computational efficient.

This paper is organized as follows. In Section 2, we will describe the material and our computational framework for drug target identification. In Sec. 3, we will describe the experimental settings and report the results. Section 4 presents the conclusions and future work.

2. MATERIALS AND METHODS

2.1 Expression Data

In this paper we used a publicly available data set comprising of experimentally acquired expression profiles[9], which is a compendium of 300 profiles of gene deletions, titratable promoter insertions and drug compound treatments. For each treatment/perturbation, a single profile was obtained from yeast cells grown to steady state after the perturbation. A log-transformed expression ratio was computed for each gene in each profile relative to untreated, wild-type yeast strains. Information regarding the identity of compounds used to treat the cells and the identity of the mutated genes in each profile was not provided to our algorithm and the test expression profile was removed from the training data set.

2.2 Regulatory Network Modeling for Drug Target Identification

In this section, we will describe our computational framework of regulatory network modeling for drug target identification. We represent a network for transcript i using a set of ordinary differential equations:

$$\dot{y_i} = f_i(y_1, ...y_N, u_i) \tag{1}$$

where $f_i(...)$ is the influence function for transcript i, y_i is the concentration of transcript i, N is the number of transcripts measured, and u_i is the net external influence on the rate of synthesis of transcript i. Since gene expression measurement technology allows only the measurement of concentrations relative to a baseline, following [4], we obtain the linear network model:

$$\sum_j a_{ij} x_j = -p_i \tag{2}$$

where a_{ij} are the model coefficients representing the influence of the concentration of transcript j on the rate of synthesis of transcript i, $x_j = log_{10}(\frac{y_j}{y_{jb}}), j \in [1..N]$ are the log-transformed expression-change ratios of each transcript, and p_i is the change ratio of the net external influences on the synthesis of transcript i. Please refer to [4] for more details.

Cross Model Building and Validation: Thus our approach can be briefly described as following:

For each compound, the following steps are processed:

- **Data Separation:** We divide our data into a training data set and a test data set similar to the splitting usually applied in cross-validation. All feature vectors belonging to the experiments related to a specific compound become the test data set, all remaining feature vectors become the training data set. In this way the model is not influenced by the compound of interest.
- **Training:** A model is then built on the training data by learning the set of coefficients of the linear combination for each gene.
- **Testing:** Once the model is trained, we apply it to expression profile of a test compound. We compute the difference between the measured expression level of each gene and the predicted level according to the level of other genes in the cell. Those genes with largest prediction errors are selected as target genes since we believe that such errors are attributed to the external influence of the compound on those genes.

More details about our approach are described in following sections.

2.2.1 *Preprocessing*

The raw data are the expression values generated by the micro-arrays. In order to compare the data from different experiments, we applied normalization so that for each x_{ij}, the value of the gene i in the experiment j, we calculate the normalized expression value

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_i}{\sigma_i} \tag{3}$$

where μ_i and σ_i are the mean and standard deviation of the expression values of gene i across all experiments, respectively.

2.2.2 *Training: Learning the Model*

Here we will describe how we learn the model given the expression data. As shown in Eq. 2, in order to learn the set of model coefficients a_{ij}, we need a set of independent experiments with measurements of x_j and the external influences p_i. However, we assume that the data are captured using perturbations for which the external influences are not known. Thus, by assuming that any treatment will directly

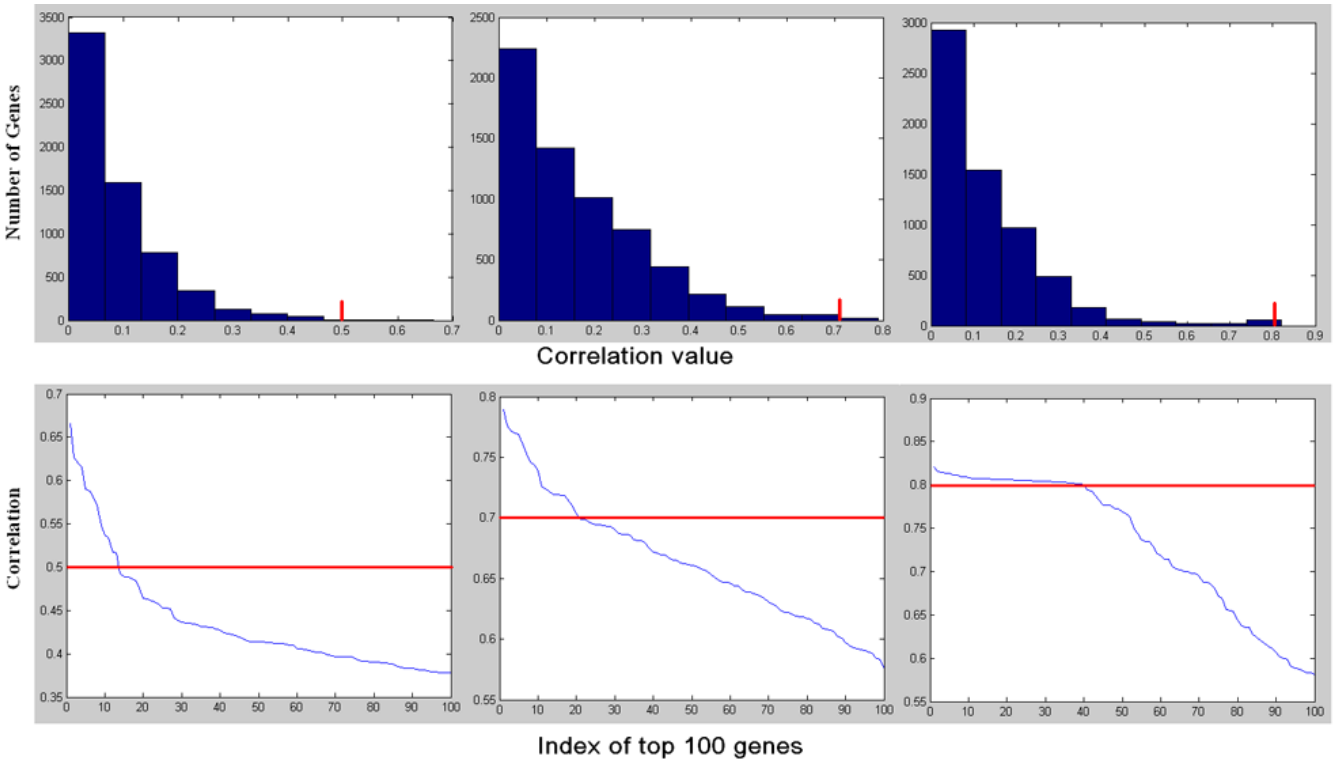

Figure 1: Correlation Analysis: for three randomly picked genes, the first row shows the distributions of the absolute values of the correlation of them and the second row plots the values of the top 100 related genes. The x-axis in the first row is the absolute value of the correlation and the y-axis is the number of genes while the x-axis in the second row is the index of top 100 genes and the y-axis is the value of correlation. We found that each gene is only highly related to a relatively small number of genes (i.e. $20-40$ genes), thus a small subset of genes can be picked for network modeling.

influence a small fraction of the genes, if we use the experiments without external influence to the gene i, p_i should be 0. Thus, using experiments without external influence, the log-transformed expression ratios of each gene can be represented by the linear combination of other genes in the network:

$$x_i = \sum_j a_{ij} x_j + d_i \tag{4}$$

where a_{ij} represent the model coefficients and d_i is the linear offset. Let $\mathbf{a}$ represent $[a_{i1}, ..., a_{iN}, d_i]^T$ and $\mathbf{x_m} = [x_{m1}, x_{m2}, ..., x_{mN}, 1]$ represents the log-transformed expression ratios of the N genes of the m-th experiments. The model coefficients a can thus be solved given a sufficient number of independent experiments as following:

$$\vec{\mathbf{x}}_i = \mathbf{Xa} \tag{5}$$

where $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_M}]^T$ is the $M \times N$ matrix with M as the number of experiments and N as the number of other genes. In order to correctly compute the coefficients, the network should have no external influences and M should be much larger than N, while typically M is much smaller than N. In [4], SVD was applied for the purpose of dimensionality reduction. Since SVD is a global dimensionality reduction method by taking account all the genes in the low dimensional space, it is impossible to have training experiments with no or small external influences on the network.

In our work, we pick a small number N' of related genes for network learning, which is both statistically and biologically reasonable since each gene is significantly influenced by a small number of other genes. Fig. 1 demonstrates the statistical validation where the first row shows the distributions of the absolute values of the correlation for three randomly picked genes and the second row plots the values of the top 100 related genes. From Fig. 1, we see that, each gene is only highly related to a relatively small number of other genes. Thus, it is better to pick a small number of genes for network learning instead of using the whole set. We applied Pearson Correlation for identifying the relationship between the genes as following:

$$c_{xy} = \frac{\frac{1}{N-1}\sum(x-\bar{x})(y-\bar{y})}{\sigma_x \sigma_y} \tag{6}$$

where x, y are two genes and $\bar{x}$ and σ_x represent the mean value and the standard deviation of gene x across all experiments respectively. For each gene, a vector of partner genes is calculated ranked by descending strength of the co-expression calculated by the correlation.

Cleaning the Training Set: In order to only use those training experiments which have no external influences to the target network, for each compound, we compute the correlations with other experiments and pick those with low correlations as the training data, more specifically, we select

those experiments ranking bottom 80% to remove feature vectors correlated with the test set. After we selected the training experiments, we can compute the model parameters for each gene according to Eq. 5.

2.2.3 Testing: Drug Target Identification

Once we estimated the network model, we can perform drug target identification for any compound. Since we have estimated the model parameters using training experiments for each gene, given test compound c, we can predict a value for each gene i:

$$x_{ci} = \sum_j a_{ci_j} x_{c_j} \tag{7}$$

The external influences can be estimated by comparing the predicted value and the true value:

$$\epsilon_{ci} = x^p_{ci} - x^t_{ci} \tag{8}$$

where x^p_{ci} represents the predicted value and x^t_{ci} represents the true value. Thus the targets of the compound c are those genes having the most significant external influences. During model learning, since the training errors are different for each gene, in order to have more accurate error estimation, we choose to use the relative errors $\epsilon^r_{ci} = \epsilon_{ci}/\epsilon^{tra}_{ci}$ where ϵ^{tra}_{ci} is the root-mean-square calculation of the training errors as shown in Eq. 9:

$$\epsilon^{tra}_{ci} = \sqrt{\frac{\sum_m \epsilon^2_{tra_m}}{M}} \tag{9}$$

where ϵ_{tra_m} is training error for the m-th training experiment and M is the total number of training experiments.

2.2.4 Two Layer Framework

In order to find the target genes for each compound, the computation for network modeling is expensive due to the large number of genes in a cell which impede the usage of comprehensive learning methods. Thus we choose to apply a two layer framework (see Fig. 2) where in the first layer we use a simple linear regression model to estimate the external influence for each gene and select a number of candidate genes for the second layer, in which a more complicated and accurate network learning algorithm is applied to achieve better results. Another advantage of using such a two layer framework is that since in the first layer, we compute the external influences across experiments for each gene, we can further remove those experiments having significant external influences from the training data for better network modeling. More specifically, in the first layer, we perform the cross model building and validation steps as described above. The Gene Targets Identification step outputs the ranked list of genes with prediction errors for each compound. We can pick a much smaller set of genes for each compound as candidate targets. Thus, instead of learning a network for each gene, we only examine those candidates, which facilitate more complicated network learning methods. In our experiments, we picked 500 genes for each compound and applied the M5 linear regression method [10] which removes the feature with the smallest standardized coefficient until there is no improvement according to the Akaike criterion[1]. The output from the first layer also provides the prediction errors of the experiments for each gene, thus we can identify those experiments which have significant errors as shown in Figure 3, which plots the prediction

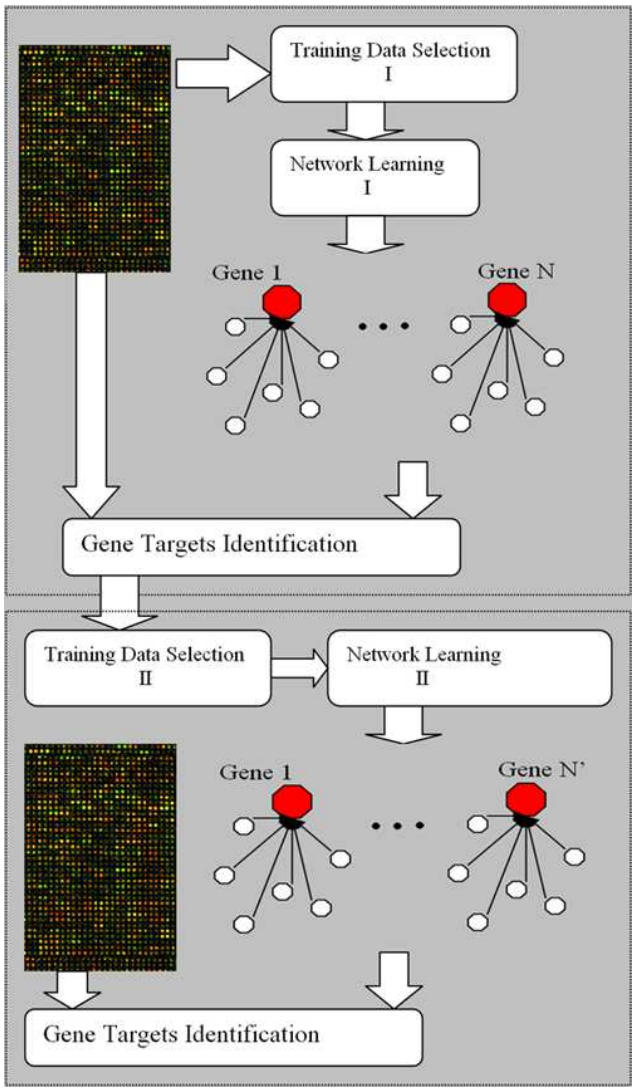

Figure 2: Two Layer Framework: In the first layer, a cross model learning and validation is applied for each gene in each experiment. The first layer provides a smaller set of genes as candidate targets for each compounds. Furthermore, prediction errors of each gene cross experiments can be used for better training data selection. More accurate network modeling and validation are performed in the second layer based on the outputs of the first layer.

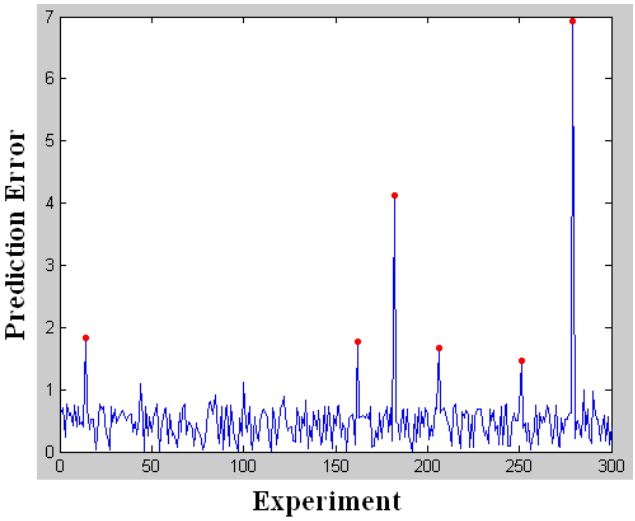

Figure 3: Error plot of gene $CDC42$: the x-axis is the index of the 300 experiments and the y-axis is the absolute values of the prediction errors. Red dots symbolize the experiments with significant prediction errors, which are unwanted peaks in the training step. From the output of the first layer, we can remove those experiments with significant errors for better network estimation.

errors of gene CDC42 cross experiments. Red dots symbolize the experiments with significant prediction errors, which are unwanted peaks in the training step. Hence, we should remove those unwanted experiments from the training set in the second layer since we aim to learn the network under no external perturbation. In Sec. 3, we will demonstrate that we achieve better results by applying the two layer framework.

3. EXPERIMENTS AND RESULTS

We performed our experiments on a publicly available, whole genome yeast expression data set which contains a compendium of 300 profiles of gene deletions, titratable promoter insertions and drug compound treatments from Hughes et al.[9]. For each treatment/perturbation, a single profile was obtained from yeast cells grown to steady state after perturbation. A log-transformed expression ratio was computed for each gene in each profile relative to untreated, wild-type yeast strains.

3.1 Network Learning Experiments

Since the drug target identification is based on the network modeling, our first set of experiments is to evaluate our network learning algorithm. It has been reported in [4] that it is very hard to identify target genes by looking at their original expression data. Fig. 4 demonstrates an example of improving drug target identification by applying a network analysis technique. In Fig. 4, the top row shows the histogram of the expression data of gene $ERG1$ of the experiments and the bottom row shows that of the prediction errors of all experiments. In both rows, the values of applying compound $Terbinafine$(targeting $ERG1$) are marked in red. By comparing the distributions of the original expression data and the prediction errors, we found that the

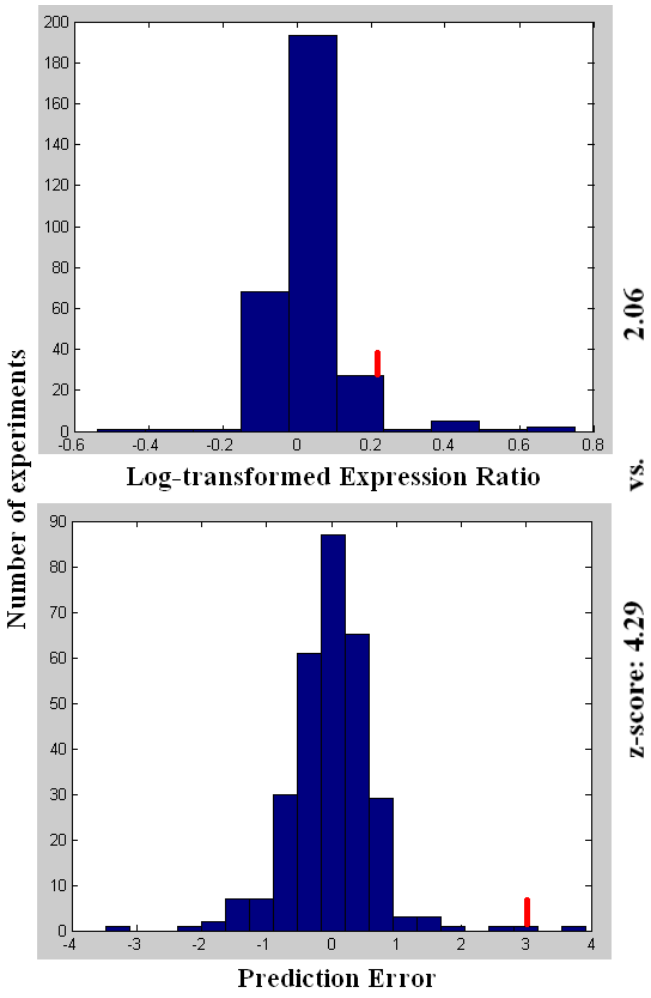

Figure 4: Histogram Comparison of gene $ERG1$: The top row shows the histogram of the expression data of gene $ERG1$ of the experiments: the x-axis represents the log-transformed expression ratios and the y-axis is the number of experiments while the bottom row shows that of the prediction errors of all experiments: the x-axis is the prediction errors. In both rows, the values of applying $Terbinafine$ are marked in red. We found that the network analysis technique improves the significance of the target gene under the applied treatment.

Table 1: Identifying targets of genetic perturbation using different numbers of network features.

Promoter mutant	Target		rank	
		10	20	30
tet-IDI1	IDI1	1	1	1
tet-RHO1	RHO1	2	2	2
tet-YEF3	YEF3	1	1	1
tet-AUR1	AUR1	1	1	1
tet-FKS1	FKS1	4	4	4
tet-KAR2	KAR2	1	1	1
tet-CDC42	CDC42	1	1	1
tet-HMG2	HMG2	4	2060	2
tet-PMA1	PMA1	138	19	17
tet-ERG11	ERG11	1	1	1
tet-CMD1	CMD1	1	1	2

network analysis technique improves the significance of the target gene under the applied treatment (i.e., z-score 4.29 vs. 2.06). Fig. 5 demonstrates more histograms of the prediction errors where the prediction errors of the target genes are statistically significant. In the next section, we will also report the target detection comparisons.

As described in Sec. 1, information regarding the identity of compounds used to treat the cells and the identity of the mutated genes in each profile should not be provided to our algorithm and the network is assumed to be learned from the data with no or very small external influences. In order to validate that the networks are correctly learned, we compare the learned networks cross experiments and we found that the variances for the learned coefficients are very small (i.e., around 5% of the mean values), which demonstrates the robustness of our approach.

3.2 Identifying Targets of Genetic Perturbation

To evaluate the performance of our algorithm, we performed the gene target identification experiments of the 11 promoter insertions. The target identification results without using the two layer framework are reported in Table 1 where the results of using different numbers of features (neighbor genes) are also compared. We found in Table 1 that the number of features significantly affects certain genes. Such a finding requires a second layer with a more comprehensive and stable network modeling method. In Table 2, we report the identification results of using the two-layer framework, which improves identification without adding much computational cost. In Table 2, we also compare our algorithm with the MNI algorithm and a simple identification algorithm (rank R) based on the z score of RNA change. The results of those methods are taken directly from [4]. We found that, our algorithm performed accurately while being efficient.

3.3 Pathway Analysis

In this section, we apply our algorithm to identify probable targets of drug compounds. Since compounds affect protein activity and only indirectly influence transcription, it is more likely to identify genes in the same pathway as the affected protein rather than the target itself. Similar

Table 2: Identification Comparison: we compare our algorithm with other methods. TLF represents our algorithm with two layer processes. MNI is method proposed in [4] and rank R is a simple identification algorithm (rank R) based on the z score of RNA change. The results of MNI and rank R methods are taken from [4] directly.

Promoter mutant	Target	TLF	MNI	rank R
tet-IDI1	IDI1	1	1	1
tet-RHO1	RHO1	2	4	1
tet-YEF3	YEF3	1	1	116
tet-AUR1	AUR1	1	1	14
tet-FKS1	FKS1	2	1	41
tet-KAR2	KAR2	1	1	64
tet-CDC42	CDC42	1	1	141
tet-HMG2	HMG2	5	1	19
tet-PMA1	PMA1	9	6	22
tet-ERG11	ERG11	1	42	2820
tet-CMD1	CMD1	1	1	1

to [4], we examine both the pathways that are represented among the highly ranked genes. Pathways are identified as significantly over-represented Gene Ontology (GO) processes among the highly ranked genes. In our experiments, instead of using the absolute value of the prediction error, we keep the sign of the error which provides us information about how the gene is affected. We examine the 6 compounds in our data set. For each of the compounds, we used our algorithm to rank more than 6000 yeast genes by the prediction errors. We then applied the highest (with highest positive errors) or lowest (with highest negative errors) ranked genes for pathway analysis using the GO Term Finder tool (http://www.geneontology.org) to identify over-represented GO biological process annotations. In Table 3, we report the most significantly selected pathway for each compound and the ranked genes in that pathway. We applied the bottom 80 genes (with highest negative errors) in our pathway analysis, which performed best. In Table 3, the first column contains the compounds, the second column shows the known pathway, the third column shows the known target genes with the rank we identified, where positive value means the rank from top(positive errors), negative value means the rank from bottom(negative values) and N/A means the target is not selected through the first layer. The fourth column shows the detected significant GO ontology with the corresponding p-value, and the fifth column shows the highly ranked pathway genes. We successfully identified the target pathway for most of the compounds we examined (5 out of 6).

In Table 4, we compare our method with other methods. The first column is the drugs, the second column is the known pathways and the third to fifth columns are the pathway analysis results of our method, MNI method and 2-fold method respectively. The MNI method is described in [4]. In the 2-fold method, for each compound, we extract a subset of genes that have 2-fold changes and put the list into the GO Term Finder tool to perform the pathway analysis. We found that, the simple 2-fold method successfully detected the target pathways for two of the compounds, but failed in other cases.

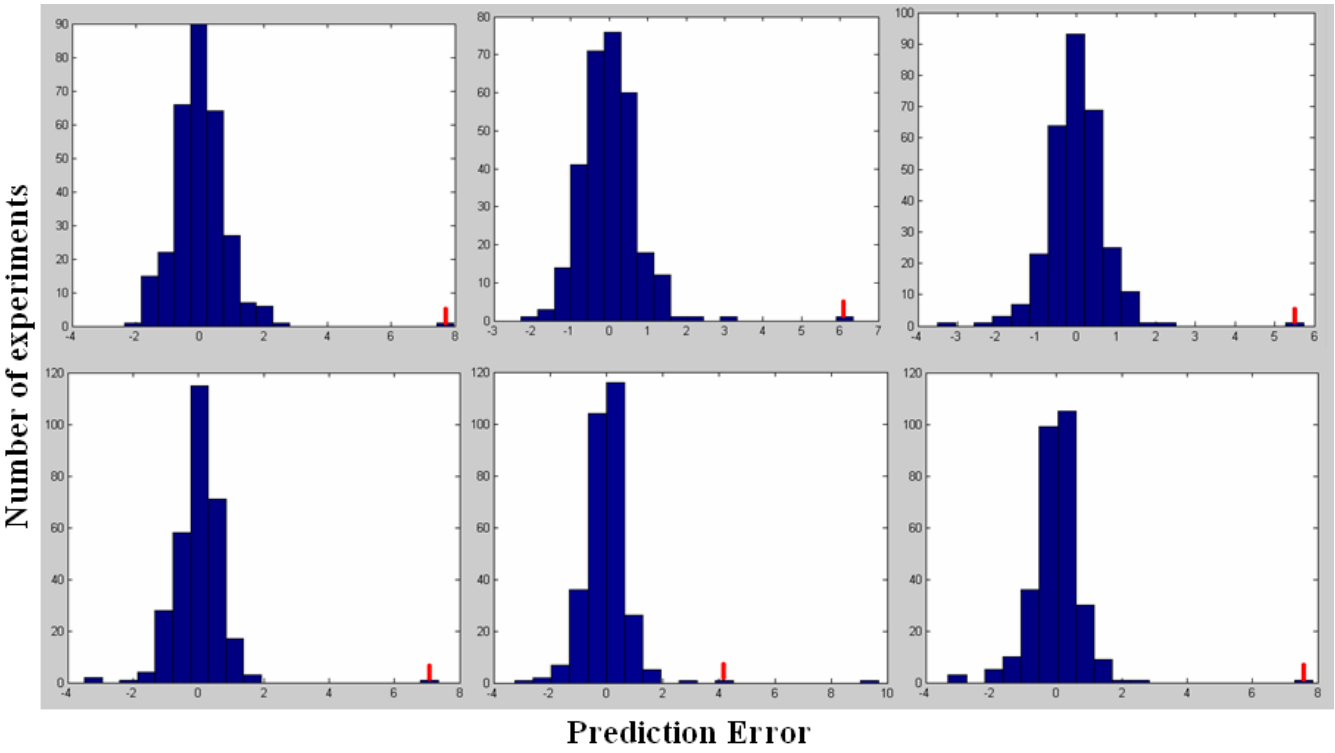

Figure 5: Histogram Analysis: The histograms of the prediction errors for 6 genes (targets of titrable promoter insertions: first row: IDI1, RHO1, YEF3; second row: AUR1, FKS1, KAR2). The titrable promoter insertion of target genes are marked in red. The prediction errors of the target genes are found to be significant by using the network analysis.

Table 3: Pathway Analysis: the first column contains the compounds, the second column shows the known pathway, the third column shows the known target genes with the rank we identified (positive value means the rank from top(positive errors), negative value means the rank from bottom(negative values) and N/A means the target is not selected through the first layer), the fourth column shows the detected significant GO ontology with the corresponding p-value, and the fifth column shows the highly ranked pathway genes.

Drug	Known Pathway	Known Target	Significant GO ontology	Ranked Pathway Genes
Terbinafine	Ergosterol Biosynthesis	ERG1(**1**)	**Steroid Metabolism** (1: 3.51e-12)	NCP1(-46), DAP1(-40), ERG7(-36) ERG8(-34), ERG26(-27), ERG12(-25) ERG2(-20), ERG28(-15), ERG24(-9) HES1(-8), ATF2(-6)
Lovastatin	Ergosterol Biosynthesis	HMG2(**-8**) HMG1(-283)	**Steroid Biosynthesis** (1: 2.96e-06)	HES1(-79), NCP1(-74), ERG7(-69) UPC2(-37), ERG11(-21), HMG2(-8)
Itraconazole	Ergosterol Biosynthesis	ERG11(N/A)	**Steroid Metabolism** (1: 0.00150)	ERG24(-34), ERG12(-17) UPC2(-13), ATF2(-6)
Cycloheximide	Protein Biosynthesis	Ribosome	**Ribosomal Protein import into Nucleus** (6: 0.00207)	NUP84(-69), NSP1(-58), NUP170(-44)
Tunicamycin	N-linked Glycosylation	ALG7(N/A)	**Protein Targeting to ER** (1: 1.58e-05)	SEC63(-71), SSS1(-42), SIL1(-38) SEC59(-35), KAR2(-12)
Nikkomycin	Cell Wall Chitin Biosynthesis	CHS3(N/A)	Response to Drug (1: 0.00062)	YKL075C(-73), YMR073C(-43) YOR129C(-27), KAP122(-10)

Table 4: Pathway Analysis Comparison: the first column contains the compounds, the second column shows the known pathway, the third column shows the pathway analysis results using our approach, the fourth column shows the results using MNI approach and the fifth column shows the pathway analysis of using the genes with a 2-fold change.

Drug	Known Pathway	Our approach	MNI approach	2-Fold
Terbinafine	Ergosterol Biosynthesis	**Steroid Metabolism** (1: 3.51e-12)	**Steroid Metabolism** (1: 10e-14)	**Steroid Metabolism** (1: 1.80e-15)
Lovastatin	Ergosterol Biosynthesis	**Steroid Biosynthesis** (1: 2.96e-06)	**Lipid Metabolism** (1: 10e-04)	**Ergosterol Biosynthesis** (1: 0.00270)
Itraconazole	Ergosterol Biosynthesis	**Steroid Metabolism** (1: 0.00150)	**Steroid Metabolism** (1: 10e-08)	No Match
Cycloheximide	Protein Biosynthesis	**Ribosomal Protein import into Nucleus** (6: 0.00207)	No Match	No Match
Tunicamycin	N-linked Glycosylation	**Protein Targeting to ER** (1: 1.58e-05)	**Protein Targeting to ER** (1: 10e-03)	No Match
Nikkomycin	Cell Wall Chitin	No Match	No Match	No Match

Table 5: Gene Selection Comparison: the first column contains the compounds, the second column shows the known pathway, the third column shows the pathway analysis results using the bottom 80 genes, the fourth column shows that of using the top 40 and bottom 40 genes and the fifth column shows that of using the top 80 genes.

Drug	Known Pathway	Bottom 80	Top and bottom 40s	Top 80
Terbinafine	Ergosterol Biosynthesis	**Steroid Metabolism** (1: 3.51e-12)	**Steroid Metabolism** (1: 8.77e-10)	**Lipid Biosynthesis** (6: 0.01311)
Lovastatin	Ergosterol Biosynthesis	**Steroid Biosynthesis** (1: 2.96e-06)	**Sterol Biosynthesis** (3: 0.00326)	Calcium-mediated Signaling (1: 0.00270)
Itraconazole	Ergosterol Biosynthesis	**Steroid Metabolism** (1: 0.00150)	**Steroid Metabolism** (1: 8.87e-05)	Amine Metabolism (1: 0.00259)
Cycloheximide	Protein Biosynthesis	**Ribosomal Protein import into Nucleus** (6: 0.00207)	Regulation of Nitrogen Metabolism (1: 0.00276)	nitrogen utilization (1: 0.00052)
Tunicamycin	N-linked Glycosylation	**Protein Targeting to ER** (1: 1.58e-05)	Response to unfolded protein (1: 0.00035)	phosphatidylserine metabolism (1: 0.00052)

Since we have both the rank and sign information, for the 5 successfully explored compounds, we compare the pathway analysis results using different genes in Table 4. We found that although the prediction error of the target gene can be positive high(ERG1 in Terbinafine), negatively high(HMG2 in Lovastatin) or not significant(ERG11 in Itraconazole), by looking at genes with significant negative errors, we can better identify the target pathways. After identifying the target pathways, the prediction errors of other genes in the target pathway may provide information about the direction of effects, i.e., for compound Terbinafine, most of the genes in the target pathway have negative prediction errors while ERG1 and ERG5 have significant positive errors. Thus one of our future research directions is to explore the directions of effects in the pathways.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a computational framework for drug target identification. In our two-layer framework, a simple linear regression model is applied in the first layer to select a small subset of candidate genes for the second layer and to refine the training data selection. A more complicate network modeling is then applied in the second layer for better results. Experimental results demonstrated that, our approach achieved accurate results in finding the known targets and associated pathways while being computational efficient.

One of our future research directions is to apply more complicated network modeling techniques in the second layer. There are two major directions, one is to apply other network modeling techniques such Bayesian Networks. Another direction is to improve the feature selection step. The M5 regression model does not guarantee global optimization, thus we aim to explore other feature selection methods for better results. We also aim to perform further validation on other data sets.

5. REFERENCES

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Autom. Control*, pages 716–723, 1974.

[2] M. Bredel and E. Jacoby. Chemogenomics: An emerging strategy for rapid target and drug discovery. *Nat. Rev. Genet.*, 5:262–275, 2004.

[3] L. Courcelle, A. Khodursky, B. Peter, and P.C. Hanawalt. Comparative gene expression profiles following uv exposure in wild-type and sos-deficient escherichia coli. *Genetics*, 158:41–64, 2001.

[4] D. di Bernardo et al. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat. Biotechnol.*, 23(3):377–383, 2005.

[5] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. In *RECOMB*, pages 127–135, 2000.

[6] T.S. Gardner, D. di Bernardo, D. Lorenz, and J.J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301:102–105, 2003.

[7] G. Giaever et al. Genomic profiling of drug sensitivities via induced haploinsufficiency. *Nat. Genet.*, 21:278–283, 1999.

[8] G. Giaever et al. Chemogenomic profiling: Identifying the functional interactions of small molecules in yeast. *Proc. Natl. Acad. USA*, 101:793–798, 2004.

[9] T.R. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

[10] Quinlan J.R. Learning with continuous classes. In *Proceedings AI*, pages 343–348. World Scientific, 1992.

[11] P.Y. Lum et al. Discoverint modes of action for therapeutic compounds using a genome-wide screen of yeast heterozygotes. *Cell*, 116:121–137, 2004.

[12] M.J. Marton et al. Drug target validation and identification of secondary drug target effects using dna microarrays. *Nat. Med.*, 4:1293–1301, 1998.

[13] G.L.G. Miklos and R. Maleszka. Microarray reality checks in the context of a complex disease. *Nat. Biotechnol.*, 22:615–621, 2004.

[14] A.B. Parsons et al. Integration of chemical-genetic and genetic interaction data links bioactive compounds to cellular target pathways. *Nat. Biotechnol.*, 22:62–69, 2004.

[15] R. Stoughton and S.H. Friend. Methods for identifying pathways of drug action. *US Patent No. 5,965,352*, 2003.

[16] J. Tegner, M.K. Yeung, J. Hasty, and J.J. Collins. Reverse engineering gene networks: Intergrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. USA*, 100:5944–5949, 2003.