A Predictive Model of Gene Expression in E. Coli

Brendan O'Donoghue, Evan Rosenfeld
Advised by Profs. Markus Covert and Daphne Koller
Project for CS 229, Stanford University, Fall 2008-09

December 10, 2008

1 Introduction

The transcription and expression of genes in organisms are primarily moderated by transcription factors. A wealth of gene expression data on E. Coli – recorded under various experimental conditions – have been collected by the scientific community. Using this data, a model was trained to learn the relationship between transcription factors and gene expression. Bonneau et. al. demonstrated that this type of analysis was possible using a smaller subset of genes on *H. salinarum NRC-1* [1]. In this paper we show that their method is generalizable to other organisms and can be extended to a nearly-complete genome. Their work was further extended by using novel machine learning techniques.

2 Motivation

Covert et. al. demonstrated that the steady-state concentrations of metabolites in a cell can be predicted to a reasonable level of accuracy using a computational metabolic network with Boolean constraints [2]. Accuracy of the predictions may be improved by removing the inflexibility of Boolean constraints and replacing them with functional constraints from our predictive model. Furthermore, the Boolean constraints were developed 'by hand' from gene expression and literature data. The method presented in this paper automates discovery of relationships between gene expression and transcription factors. These relationships can be used to replace or augment Boolean constraints.

3 Methods

Gene expression data was collected for approximately 4500 genes across approximately 1000 experimental conditions from the University of Oklahoma E. Coli Gene Expression Database [3].

3.1 Data Preprocessing

Data preprocessing is an important step which aggregates the data into a standard format and corrects for data anomalies which could confound our findings. A PHP script was written to correct for common data-entry mistakes such as frame-shift errors and to prepare the data to be imported into MATLAB. E. Coli transcription factors were identified from a separate database. The transcription factors were removed from the dataset and placed in a separate matrix. Experiments or genes that were missing more than 10% of their total data were removed from the dataset so that they would not bias the biclusters. It was necessary to normalize the data along the experiment dimension because different technicians and different labs introduce significant variation into the collected data. Normalization is an attempt to remove some of this variation.

3.2 Biclustering

Initially, hierarchical K-means clustering was run on the dataset to identify co-expressed genes along a subset of experimental conditions. It was later decided that co-expressed genes are not necessarily co-regulated. As this paper seeks to explore transcription factor regulation of genes, a more advanced technique was required to identify co-regulated genes.

Biclustering is a method for simultaneously clustering along the rows and columns of a matrix to find highly correlated subsets of rows and columns within a dataset. Church and Cheng have proposed that biclustering is a more biologically relevant form of clustering [4] for co-regulated network discovery. The variation of biclustering employed in this paper worked as follows:

1. A random subset of genes and experimental conditions is chosen as a seed bicluster

2. Let B be the set of rows in the bicluster. The variances of all columns in the dataset and the means of all columns using only the rows in B are found:

$$\mu_j = \frac{1}{I}\sum_{i=1}^{I} M_{ij}$$

$$\sigma_j^2 = \frac{1}{I}\sum_{i=1}^{I}(M_{ij} - \mu_j)^2$$

$$\mu_{j'} = \frac{1}{B}\sum_{i \in B} M_{ij}$$

3. A Z-score is calculated for every element in the dataset: $Z_{ij} = \frac{M_{ij} - \mu_j^{'}}{\sigma_j}$.

4. Z-scores are transformed into probabilities $p(x_{ij}|B)$ by integrating the two tails above and below $\pm|Z_{ij}|$ of a standard normal distribution. The values represent the probability of each element being in the bicluster given the current bicluster composition.

5. Elementwise products are taken across rows and columns to generate the probabilities of the rows and columns being in the bicluster.

$$p(x_j|B) = \prod_i p(x_{ij}|B)$$

$$p(x_i|B) = \prod_j p(x_{ij}|B)$$

6. These probabilities are compared to a random number to select rows and columns to be included into the bicluster based on an annealing schedule.

7. Steps 2 to 6 are repeated until convergence: when the rows and columns of the bicluster do not change.

The biclustered genes traces are then averaged to produce one signal to be regressed on. This reduces the noise of the signal to be regressed, as co-regulated genes should have the same expression level under their co-regulation conditions.

3.3 Generating Regressors

In order to reduce the dimensionality of the problem the transcription factors were clustered using K-means clustering. Within K-means, the Pearson correlation coefficient distance metric was used instead of the more common ℓ_2 norm. This is because the goal is to cluster correlated genes. Two genes that are perfectly correlated but scaled may have a larger ℓ_2 distance than two genes that are not as well correlated but of similar range.

It was desirable to cluster highly correlated genes because the regressors generated from this process were used in an ℓ_1 regularized regression. If two genes are perfectly correlated but scaled versions of each other, the ℓ_1 regularization will pick out only the larger one because it can attach to it a smaller coefficient. Therefore, we would not identify a relationship between the smaller scaled gene and the signal. By using the Pearson correlation coefficient distance metric, we have a higher probability of identifying correlated but scaled genes as important regressors of the signal.

The transcription factor signals in the clusters are then averaged.

Next pairwise minimums of all of the cluster traces were generated. The minimum of two continuous functions is the continuous analog of the Boolean AND of two binary signals. By incorporating the 'AND's and the signals themselves, the continuous analogs of OR and XOR can also be generated as shown in the table below [5]:

	AND	OR	XOR
min(a,b)	1	-1	-2
a	0	1	1
b	0	1	1

There is a problem that arises from this table. Because the ℓ_1 regularization penalizes each coefficient, XORs and ORs are more expensive to include than ANDs. An attempt was made to incorporate ANDs, ORs and XORs as their own variables in order to solve this problem, but this process was too computationally intensive.

3.4 Regression

In order to test our regression models, holdout cross validation was employed. Regression was performed on 60% of the bicluster averages, and the remaining 40% was used to test the regression model. Regressions were performed in two steps:

1. A standard least-squares regression was performed to identify a parameter vector β_{unc}

$$\text{minimize} \quad \|R\beta_{unc} - s\|_2 \ ,$$

where s is the training subset of the bicluster average and R is the training subset of the matrix of regressors.

2. A least-squares regression with ℓ_1-regularization is performed:

$$\begin{array}{ll} \text{minimize} & \|R\beta_c - s\|_2 \\ \text{subject to} & \|\beta_c\|_1 \leq t\|\beta_{unc}\|_1 \end{array}$$

These objectives are all convex and as such can be easily solved with a standard convex solver [6, 7]. At the end of this process, with the appropriate value for t , only the regressors that best explain the data will be active. It is

possible to perform a third regression on these active regressors without a constraint term to find the optimal values for the active regressors, this is called polishing. However, polishing was not found to significantly improve the results.

Thus $R\beta_c = \hat{s}$ becomes our best fit estimate of the signal, s.

3.5 Measures of Goodness

Three measures of goodness were used to evaluate the accuracy of the regressions on the test set.

1. $\eta = \frac{\|s-\hat{s}\|_2^2}{\|s\|_2^2}$: a normalized squared error ratio. When $0 \leq \eta < 1$, the regression predicts the signal to some degree.

2. $\rho(s,\hat{s})$: the standard cross-correlation between s and $\hat{s}$.

3. $MSE = \frac{1}{N}\|s-\hat{s}\|_2^2$: the mean-squared error of the regression.

4 Results

After preprocessing, the dataset comprised of 3638 genes across 664 experimental conditions and 157 transcription factors. K-means clustering resulted in approximately 130 clusters of transcription factors, depending on the particular bicluster. These were converted into approximately $8,500$ regressors, including a constant regressor, which was not subject to the ℓ_1 regularization.

An example bicluster with 10 genes across 100 conditions is shown in figure 1. As you can see, the biclustering algorithm has selected a highly correlated subset of genes across a subset of conditions. The genes selected by this bicluster were yeiM, hyfE, hyfG, pbpC, ygdB, ygeX, hofQ, sgbH, ulaG, and ulaA. These genes show an enrichment in E. Coli. energy metabolism and transport, although many were not yet classified. The results of the regression selected several transcription factors that were suspected regulators of many of the biclustered genes, as well as new ones that may play an active role not yet discovered experimentally.

It was noted that many biclusters were similar, even with different random starting conditions. This is due to the fact that many genes and conditions are quite dominant and are repeatedly selected by the algorithm.

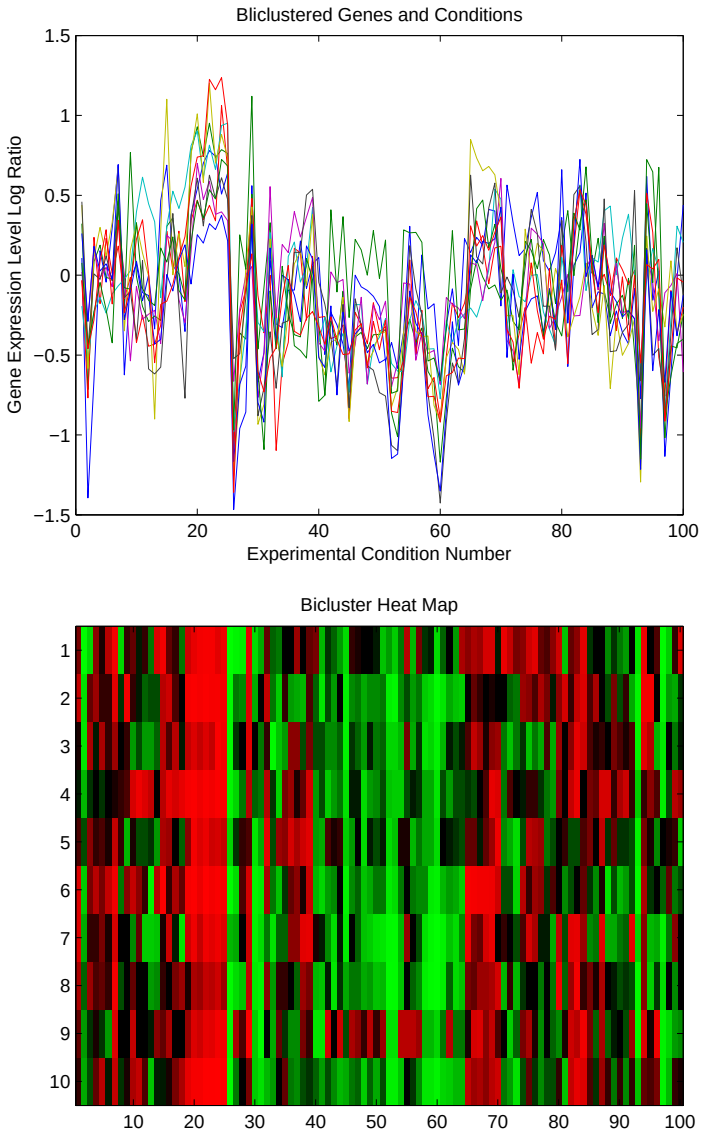
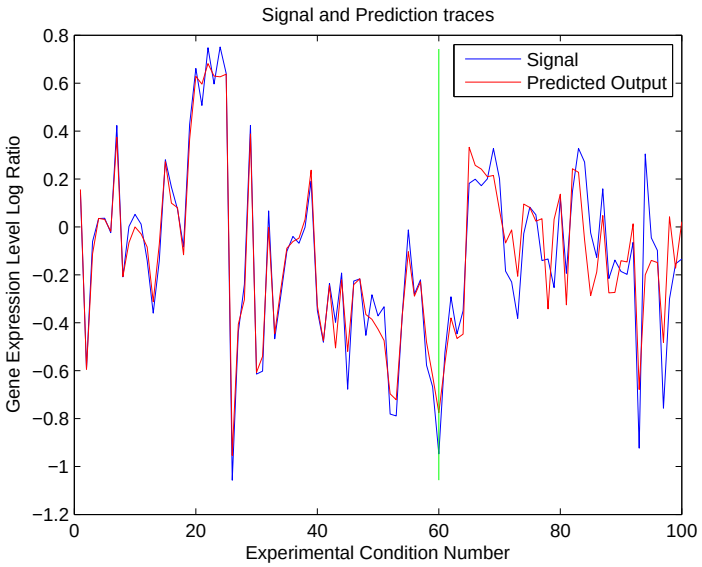


Figure 1: Example Bicluster with 10 genes across 100 conditions.

Figure 2 shows the results of the regressions on the training and testing signals of the bicluster in Figure 1 with less than 0.4% of the regressors active. In the top subfigure, samples to the left of the green line are the training set, samples to the right are the test set. The regression typically produced excellent results, for this regression in particular, $\eta = 0.579$, $\rho = .796$, and MSE $= 0.0287$.

Overfitting of the training data is always a concern in learning applications. Figure 3 shows the bias/variance trade-off as a function of the regularization constant, t. As t increases, the training error decreases to zero as the variance of the regression predictions overfit the training signal. At the same time, the testing error initially decreases as the regression learns the underlying relationships between the regressors and the training signal, but then the testing error increases as the high variance of the prediction de-



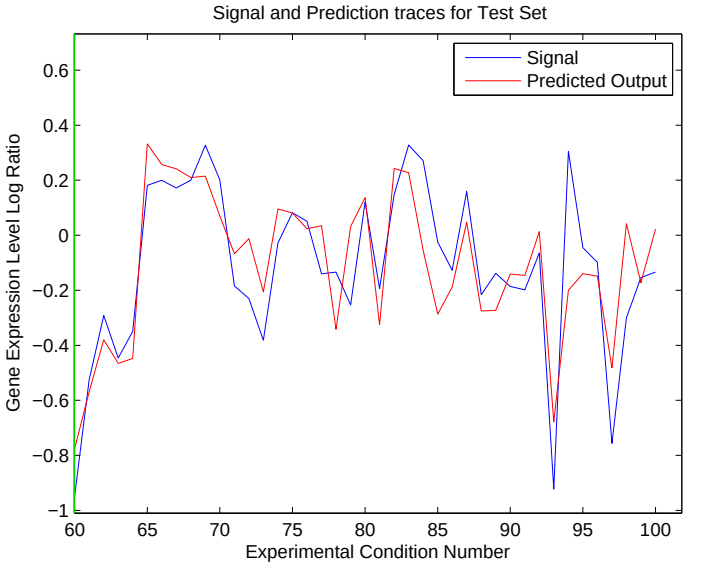


Figure 2: Regression onto Training (Left) and Testing Signal [above], Close-up of Testing Signal [below]

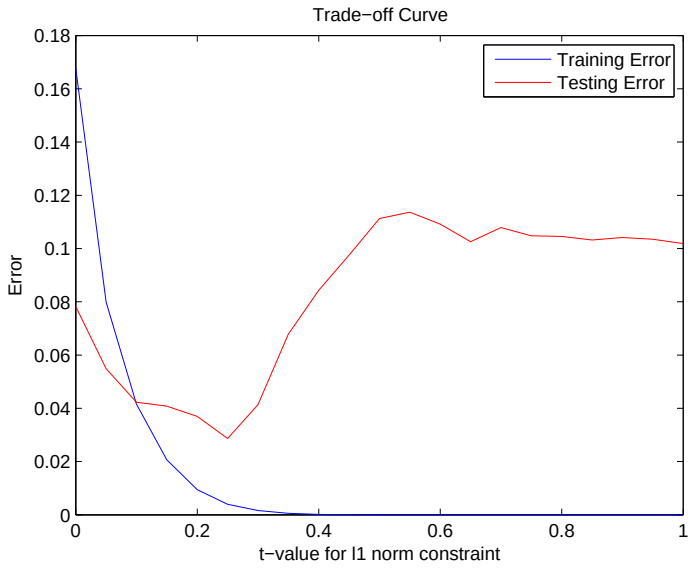


Figure 3: Bias/Variance trade-off curve

creases the ability to generalize. There is an optimal point where the training data is neither overfit nor underfit, and the testing error is minimized. This occurs around 0.25 for this bicluster.

Figure 4 shows some other bicluster signal and predicted outputs. By comparing to Figure 2 we observe the dominance of certain genes and conditions that appear in many independent biclusters.

5 Conclusions

In this paper, we have demonstrated that it is possible to take freely available data and to generate an accurate predictive model of gene expression using only transcription factor data. Many of the genes in E. Coli. (and other organisms) have not yet been classified. The methods presented in this paper can provide clues for gene function classification both by the bicluster a gene belongs to and the predicted output of expression data under various conditions.

On top of this, the functional relationships that we have discovered between gene expression data and transcription factors can potentially be used to replace the Boolean constraints in the E. Coli metabolic network of Covert et. al[2]. We have also shown that Bonneau's work on *H. Salinarum NRC-1* can be performed on nearly-complete genomes of other organisms.

6 Further Work

The methods presented in this paper are a good first step to explain expression data. There are several other consid-

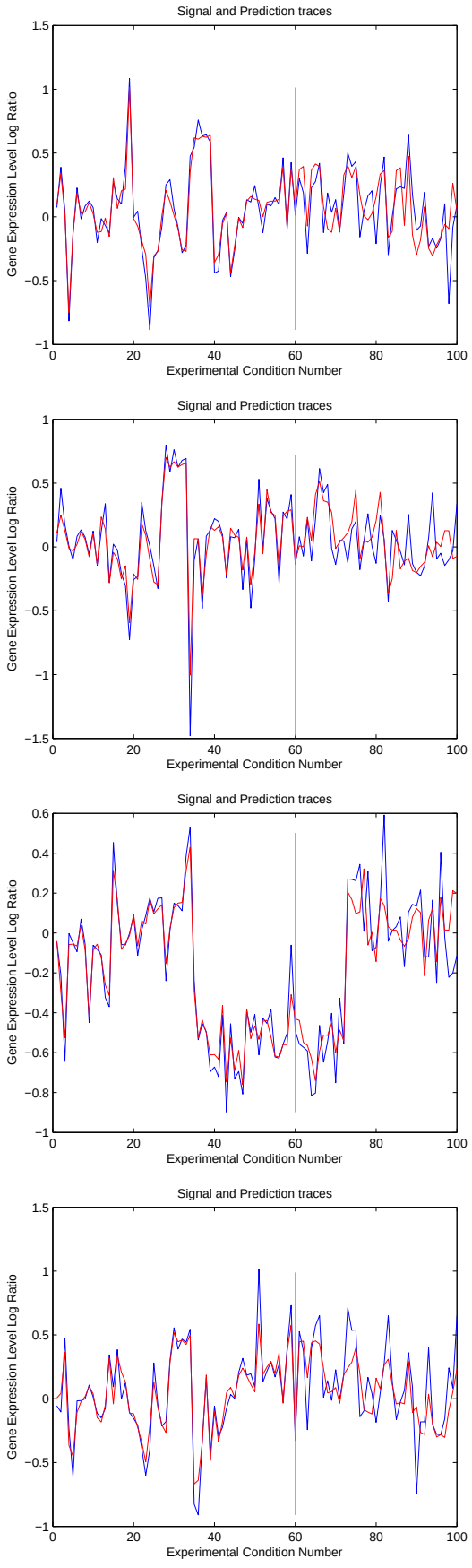


Figure 4: Signal and prediction traces for various biclusters, the blue trace is the signal and the red trace is the prediction. Samples to the right of the green line are the validation set.

erations that would extend this work.

In order to improve the accuracy of the biclustering algorithm, transcription factor binding motif data and priors based on other network connectivity models could be incorporated. A penalty factor could also be incorporated to prevent dominant genes and conditions from repeatedly entering biclusters.

The AND, XOR, and OR signals could be represented as single regressors to remove the ℓ_1 bias that makes XOR and OR more expensive to incorporate into regression models than AND.

It is important to consider what factors are regulating the transcription factors. It is possible that one transcription factor is regulating another, or both are being co-regulated by a third, confounding transcription factor. Due to these uncertainties, the causality of predictions generated using the model in this paper should be examined further.

Transcription factors are not the sole regulators of gene expression, environmental factors play a strong role too. Any full predictive model would have to incorporate these data to be accurate and robust.

Finally, it is important to experimentally validate any predictions generated by this model.

References

[1] R. Bonneau, et. al (2007). A Predictive Model for Transcriptional Control of Physiology in a Free Living Cell. Cell 131, pp. 1354-1365.

[2] M. Covert et. al, Regulation of Gene Expression in Flux Balance Models of Metabolism (Journal of Theoretical Biology 213:73-88, 2001).

[3] Oklahoma University E. Coli Gene Expression Database. Available: http://genexpdb.ou.edu/.

[4] Cheng Y, Church GM (2000). Biclustering of expression data. Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology: 93–103.

[5] R. Bonneau, et. al (2006). The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. Genome Biology 7:R36

[6] S. Boyd, L. Vandenberghe, Convex Optimization (Cambridge Univ. Press, 2004).

[7] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming (web page and software). http://stanford.edu/~boyd/cvx, May 2008.