Identifying and Understanding Differential Transcriptor Binding

David Koes and Yong Lu
{dkoes,lyongu}@cs.cmu.edu

April 30, 2004

Abstract

We study differences in transcription factor binding under different conditions as a classification problem. We use several simple classifiers to predict binding under a rapamycin condition in yeast cells based on expression data, protein-protein interaction data, and binding data. The classifiers are very good at fitting existing data and replacing missing values, do fairly well at predicting the results of a single binding experiment given the results of several others, and may be a useful tool for guiding experimentation, but is not clear that they can provide underlying biological insight as to why transcription factors bind differently under different conditions.

1 Introduction

Transcription factors, proteins that bind to DNA and regulate the expression of genes, bind to different sets of genes under different conditions. Understanding when and why transcription factors change binding preferences would provide valuable insight into the underlying metabolic pathways responsible for the cell's response to changing conditions. In this paper we study differences in transcription factor binding caused by exposure to the cytotoxin rapamycin. We attempt to answer the following questions:

- Can we fill in the missing values of a binding experiment?
- Can we predict all the values for a single binding experiment?
- Can we use our results to suggest useful additional experiments to perform?
- Can we explain the differences in binding?

In order to answer these questions we treat determining binding under rapamycin as a classification problem. The classifier takes as input a transcription factor/gene pair and, using features specific to this pair, classifies the pair as either binding under the rapamycin condition or not. We investigate using several types of classifiers and sets of features.

2 Data Sources

We use several different data sources to generate the training and test data for our classifiers. All data is for *Saccharomyces cerevisiae* (yeast). We use the micro-array expression data available from `http://www-schreiber.chem.harvard.edu/home/protocols/partitioning/` which provides expression ratios for both YPD and rapamycin conditions. This data indicates what genes are currently being transcribed and may also serve as an indirect indicator of protein levels. We use the genome wide location analysis data from `http://web.wi.mit.edu/young/regulator\_network/` (YPD) and `http://www.psrg.lcs.mit.edu/Networks/modules.html` (rapamycin). This data provides p-values that indicate the likelihood that a transcription factor is

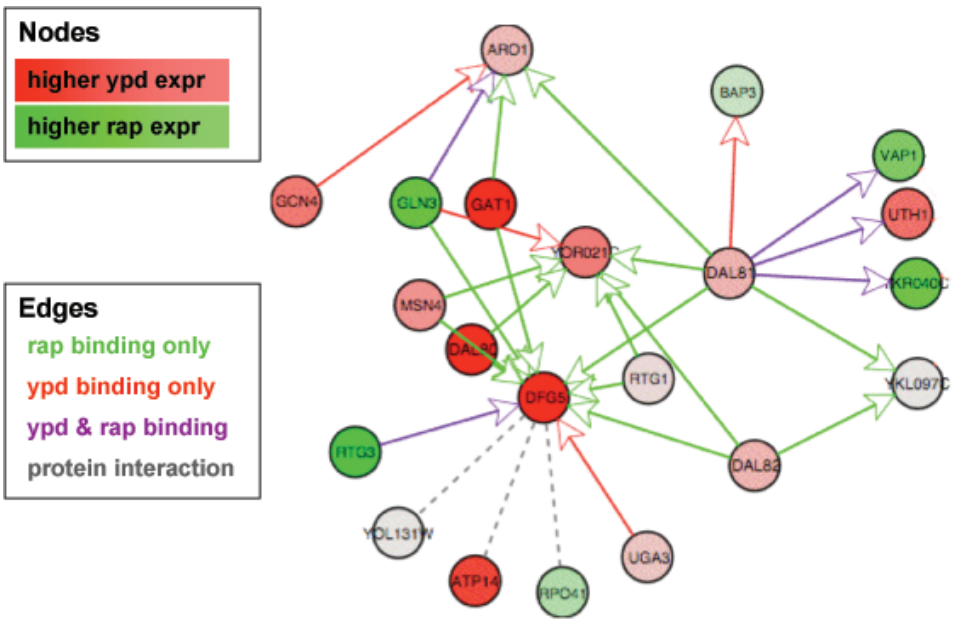

Figure 1: A small subset of a difference graph between YPD and rapamycin conditions. Notice that the transcription factor DAL81 stops, starts, and continues binding to various genes.

binding to a gene. The YPD data contains data for 106 transcription factors while the rapamycin data only has information for 14 transcription factors which were considered to play a role in the cell's response to rapamycin. The datasets share 12 transcription factors in common. Finally, we used protein-protein interaction data generated using the two-hybrid method from `http://genome.c.kanazawa-u.ac.jp/Y2H/`.

When combined, the various datasets form a network of biological interactions under two conditions. This network can be represented as a difference graph as shown in Figure 1.

3 Feature Selection

In order for the data to be useful to a classifier, features that are capable of distinguishing binding transcription factor/gene pairs must be extracted from it. That is, the features cannot be global features (such as the fact that a specific gene has a certain expression level) but must relate to the potential binding pair (such as the expression level of the transcription factor). We therefore use the expression and binding data for genes that are "close" to the potential binding pair in the network formed from the protein-protein and protein-dna (binding) data. The intention is for this network to be defined by the *capabilities* of the molecules in question. It is not necessary for a factor to be observed binding to another gene under the conditions being studied, it would be sufficient to know that the factor is capable of binding under some condition. However, in this study we use only the YPD and rapamycin data to determine binding capabilities.

We study both a sparse and a dense representation of features.

- **Sparse Representation** Figure 2. This representation has several attributes for every gene, but only assigns non-zero values to those genes which are "close" to the binding pair in question. Each gene has the following attributes:
  - `GENE_binds_factor_ypd` binding p-value for gene with the factor
  - `GENE_binds_target_ypd` binding p-value for gene with the target

Nonzero attributes (16)

factor_binds_target_rap	**.6**
factor_binds_target_ypd	.001
factor_binds_C_ypd	.001
B_binds_target_ypd	.002
factor_self_A_ypd	2.0
factor_ldown_C_ypd	-2.0
factor_lpp_B_ypd	-1.0
factor_self_A_rap	1.0
factor_ldown_C_rap	2.0
factor_lpp_B_rap	1.0
target_self_C_ypd	-2.0
target_lup_A_ypd	2.0
target_lup_B_ypd	-1.0
target_self_C_rap	2.0
target_lup_A_rap	1.0
target_lup_B_rap	1.0

Zero attributes (46)

factor_binds_A_ypd
factor_binds_B_ypd
target_binds_A_ypd
target_binds_B_ypd
<etc.>

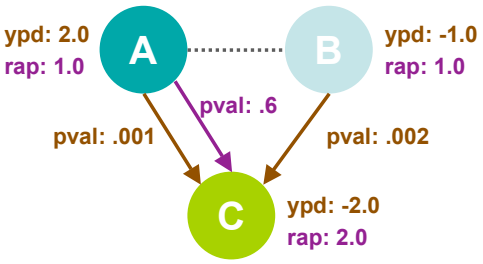

Figure 2: A simple example of the sparse features for the potential binding pair **AC**. `factor_binds_target_ypd` is what is being predicted. It is present, in binary form, in the training data but not in the test data.

Nonzero attributes (12)

rap_bind	**.6**
ypd_bind	0.001
factor_expr_ypd	2.0
factor_expr_rap	1.0
target_expr_ypd	-2.0
target_expr_rap	2.0
target_ave_expr_up_YPD	**0.5**
target_ave_expr_up_RAP	**1.0**
factor_ave_expr_down_YPD	-2.0
factor_ave_expr_down_RAP	2.0
factor_ave_expr_pp_YPD	-1.0
factor_ave_expr_pp_RAP	1.0

Zero attributes (6)

target_ave_expr_down_YPD	0
target_ave_expr_down_RAP	0
target_ave_expr_pp_YPD	0
target_ave_expr_pp_RAP	0
factor_ave_expr_up_YPD	0
factor_ave_expr_up_RAP	0

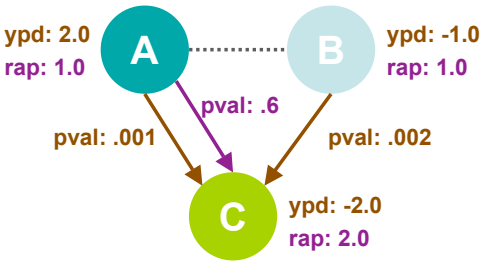

Figure 3: A simple example of the dense features for the potential binding pair **AC**. `rap_bind` is what is being predicted. Notice that the values for `target_ave_expr_up_YPD` and `target_ave_expr_up_RAP` are the averages of the expression values for **A** and **B**.

- factor_1up_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if it can bind the factor
- target_1up_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if it can bind the target
- factor_1down_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if it can be bound by the factor
- target_1down_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if it can be bound by the target
- factor_1pp_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if a protein-protein interaction exists between its product and the factor
- target_1pp_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if a protein-protein interaction exists between its product and the target's
- factor_self_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if it is the factor
- target_self_GENE_<ypd|rap> expression under YPD or rapamycin condition of gene if it is the target

- **Dense Representation** Figure 3. This representation uses a small constant number of attributes for each binding pair by only taking the average of expression levels that are "close" to the binding pair in a specific direction resulting in the following 17 attributes:
  - ypd_bind binding p-value of factor and target under YPD conditions
  - factor_expr_<YPD|RAP> expression under YPD or rapamycin conditions of factor
  - target_expr_<YPD|RAP> expression under YPD or rapamycin conditions of target
  - factor_ave_expr_up_<YPD|RAP> average expression under YPD or rapamycin conditions of genes that bind to the factor
  - target_ave_expr_up_<YPD|RAP> average expression under YPD or rapamycin conditions of genes that bind to the target
  - factor_ave_expr_down_<YPD|RAP> average expression under YPD or rapamycin conditions of genes that are bound by the factor
  - target_ave_expr_down_<YPD|RAP> average expression under YPD or rapamycin conditions of genes that are bound by the target
  - factor_ave_expr_pp_<YPD|RAP> average expression under YPD or rapamycin conditions of genes whose protein interacts with the factor protein
  - target_ave_expr_pp_<YPD|RAP> average expression under YPD or rapamycin conditions of genes whose protein interacts with the target

Both the sparse and dense feature representations have their advantages and disadvantages. The sparse feature set precisely captures the available data and provides per-gene information to the classifier. Although most of the features are zero, allowing for a compact representation, the number of attributes is proportional to the number of genes resulting in slower classification. The preciseness of the representation may also lend itself to overfitting the data resulting in poorer classifications. Finally, the classifiers we use require that sparse feature representations be binary, consisting of only zeroes and ones. That is, it is necessary for us to discretize the data. We do so by breaking up expression values into hi and lo variables. A gene's expression is considered high if it is positive and low if it is negative. For binding values, a p-value below .01 is considered to imply binding.

The dense data set has a fixed number of attributes for each training example resulting in fast classification. In addition, there is little overhead in adding additional attributes should that be considered desirable. The classifiers that support dense input data do not require binary values and so can fit the classification to the actual expression and binding values. However, because of the averaging of expression values this representation does not capture as much information as the sparse representation.

4 Classifiers

We used three different types of classifiers:

- naiïve Bayes
- K Nearest Neighbors
- Logistic Regression

We used the Auton Fast Classifier software available from the Auton Lab (`http://www.autonlab.org`).

4.1 naiïve Bayes

This method makes the simplifying assumption that the attribute values are conditionally independent given the classification output value. It computes the probability of binding given a set of training attributes by:

$$P(y = BIND|x_1, \ldots x_m) =$$
$$\frac{P(x_1, \ldots x_m|y = BIND)P(y = BIND)}{P(x_1, \ldots x_m|y = BIND)P(y = BIND) + P(x_1, \ldots x_m|y = NOT\_BIND)(1 - P(y = BIND))}$$

By making the simplifying assumption that the attributes are conditionally independent given the output value, this can be simplified to:

$$\frac{P(y = BIND)\prod_{k=1}^{m} P(x_k|y = BIND)}{P(y = BIND)\prod_{k=1}^{m} P(x_k|y = BIND) + P(y = NOT\_BIND)\prod_{k=1}^{m} P(x_k|y = NOT\_BIND)}$$

The conditional probabilities, $P(x_k = v|y = BIND)$, can be readily estimated from the training data simply by counting the number of training examples with $x_k = v$ and $y = BIND$ and dividing by the total number of examples where $y = BIND$.

This method has the advantage of being very fast. Unfortunately the simplifying assumption is unlikely to hold in this context. Furthermore the algorithm must discretize the data in order to work.

4.2 K Nearest Neighbors

In this classifier a test example is classified by identifying the k nearest training examples in attribute space. The test example is then classified based upon what fraction of these examples demonstrate binding in the rapamycin condition. For our experiments we used a value of 9 for k. This classifier can readily deal with continuous valued attributes but will only classify well if identically classified examples are, in fact, clustered together in the attribute space.

4.3 Logistic Regression

In Logistic Regression, the classifier finds the parameter β such that the function

$$u_i = \frac{e^{\beta \cdot \mathbf{x}_i}}{1 + e^{\beta \cdot \mathbf{x}_i}}$$

maximizes the log-likelihood with the classification result y_i:

$$y_i ln(u_i) + (1 - y_i) ln(1 - u_i)$$

Standard gradient optimization techniques are used. Like k-nearest neighbors, this method can also readily handle continuous valued attributes but can perform more complex classifications.

5 Results

We find that the classifiers are very good at fitting existing data and replacing missing values, do fairly well at predicting the results of a single binding experiment given the results of several others, and may be a useful tool for guiding experimentation, but it is not clear that they can provide an underlying biological insight as to why transcription factors bind differently under different conditions.

5.1 Missing Value Prediction

In any high throughput experiment it is likely there will be missing values relatively randomly distributed throughout the dataset. Rather than ignoring these value or running additional experiments, it would be preferable to attempt to predict these values based on the known values. In order to evaluate how well the various classifiers would perform at this task we perform k-fold validation. In k-fold validation a randomly choosen $1/k$th of the data is used as test data while the remaining data is used to train the classifier. Our dataset contains all YPD/rapamycin binding pairs for the 12 transcription factors that are present in both the YPD and rapamycin binding datasets. In addition we include approximately three times as many randomly chosen binding pairs which are known not to bind in either condition. Including these pairs is necessary to avoid overfitting the data (for example, without them all pairs that do not bind in the YPD condition would bind in the rapamycin condition). It is computationally prohibitive to include all the possible binding pairs for the 12 transcription factors.

Results are presented in the form of ROC curves. An ROC curve displays how the false positive fraction (the horizontal axis) and true positive fraction (the vertical axis) change as the threshold of acceptance is modified. That is, each classifier produces a probability that a test example will bind under the rapamycin condition. As we accept lower probabilities as still indicating the occurrence of binding we expect both the number true and false positives to increase. In a good classifier there will exist a threshold for which there are many true positives and few false positives. This results in an ROC curve which has a steep initial slope and a very sharp knee. It is easy to see that as the area under the ROC curve approaches one, the accuracy of the classifier approaches 100%. Conversely, a classifier that randomly classified over a uniform distribution would appear as a straight line and have an area under the curve of .5.

The ROC curves for the three classifiers and two feature representations are shown in Figure 4. All do better than a random classifier with the logistic regression and k-nearest neighbor classifiers outperforming the naiïve Bayes classifier. As expected, the accuracy improves as the number of folds increases as the test set becomes correspondingly smaller and there is more training data. Somewhat surprisingly, as much as a third of the data can be missing without significantly affecting the accuracy of the classifier in most cases. Note that the sharp knee in the k-nearest neighbor curve is an artifact of the relatively small value of k neighbors we look at (9) which results in only 9 possible probabilities as output from the classifier. As a summary of these results, the area under the ROC curves is plotted in Figure 5.

If our goal is simply to predict a few missing values within a binding experiment, it may not necessarily be beneficial to use data from multiple experiments. Instead, the classifier might do a better job if only the data from a single transcription factor binding experiment is used. To test this hypothesis we ran an 8-fold validation experiment for each of the transcription factors. The results are shown in Figure 6. In most cases the classifiers can do better with the more relevant, if smaller, datasets. Logistic regression using the sparse dataset, a combination that scores the best overall, can almost perfectly predict the results.

5.2 Experiment Prediction

Although predicting missing values is useful and demonstrates the potential accuracy of the classifiers, a more interesting and challenging application is using the classifiers to predict the results of a new binding experiment using a different transcription factor. In order to evaluate the feasibility of such an application, for each of our 12 transcription factors we leave one factor out, train the classifier with 11 transcription factors and then use the left out data as the test set.

The ROC curves for these experiments are shown in Figure 7 with the areas under the curves displayed in Figure 8. The naiïve Bayes approach with the sparse feature representation performs at essentially the same level as a random

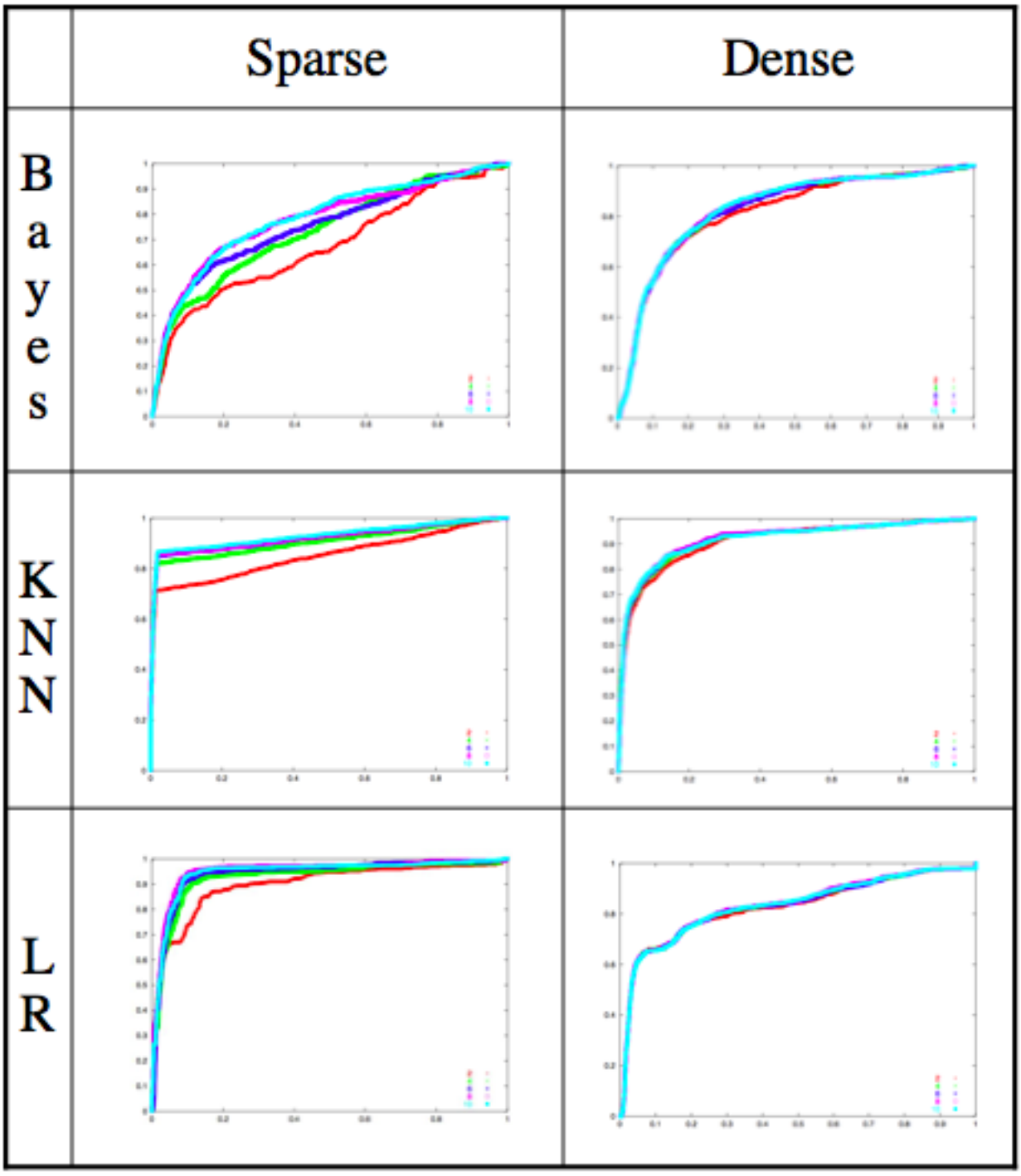

Figure 4: ROC curves for the various classifiers and feature representations with k-folds validation for multiple values of k. All do better as k increases (resulting in a smaller test set). Logistic regression and k-nearest neighbors clearly outperform the naiïve Bayes approach and all do better than a random classifer.

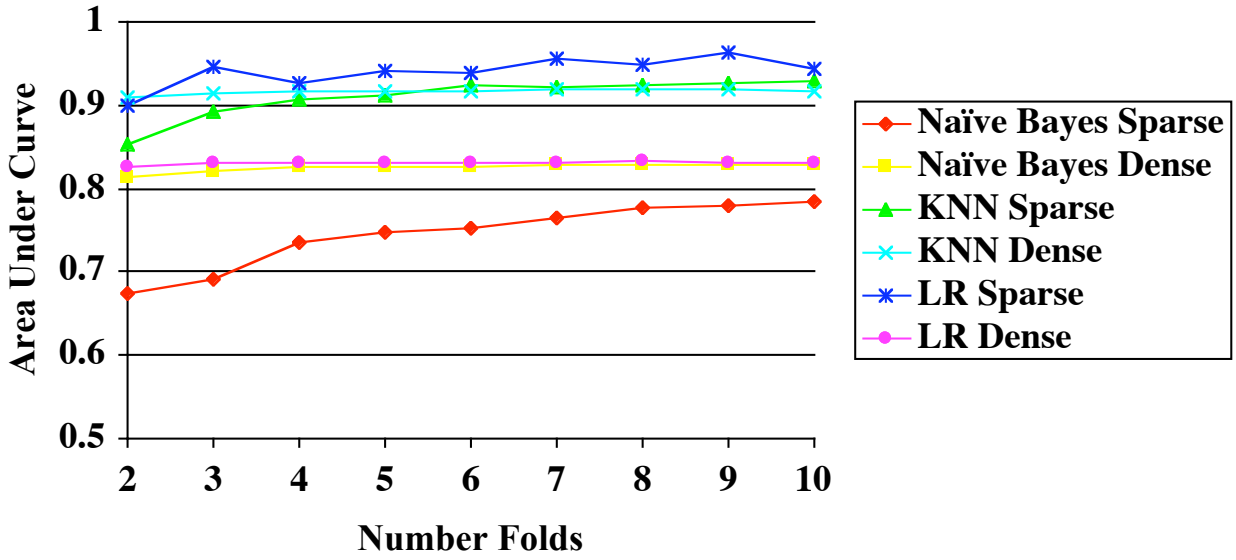

Figure 5: The area under the ROC curves for the various classifiers and feature representations as the number of folds in k-folds validation is increased.

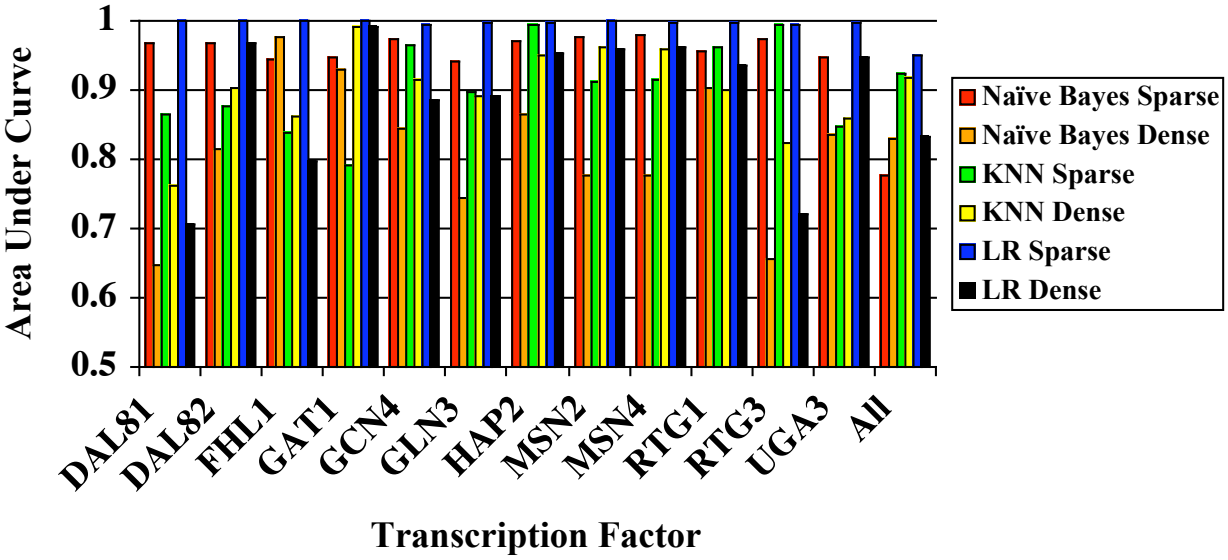

Figure 6: The area under the ROC curves for the various classifiers and feature representations for 8-fold validation using just the data for each individual transcription factor. In most cases the classifiers can actually do better with less data. Logistic regression using the sparse feature representation does almost perfectly.

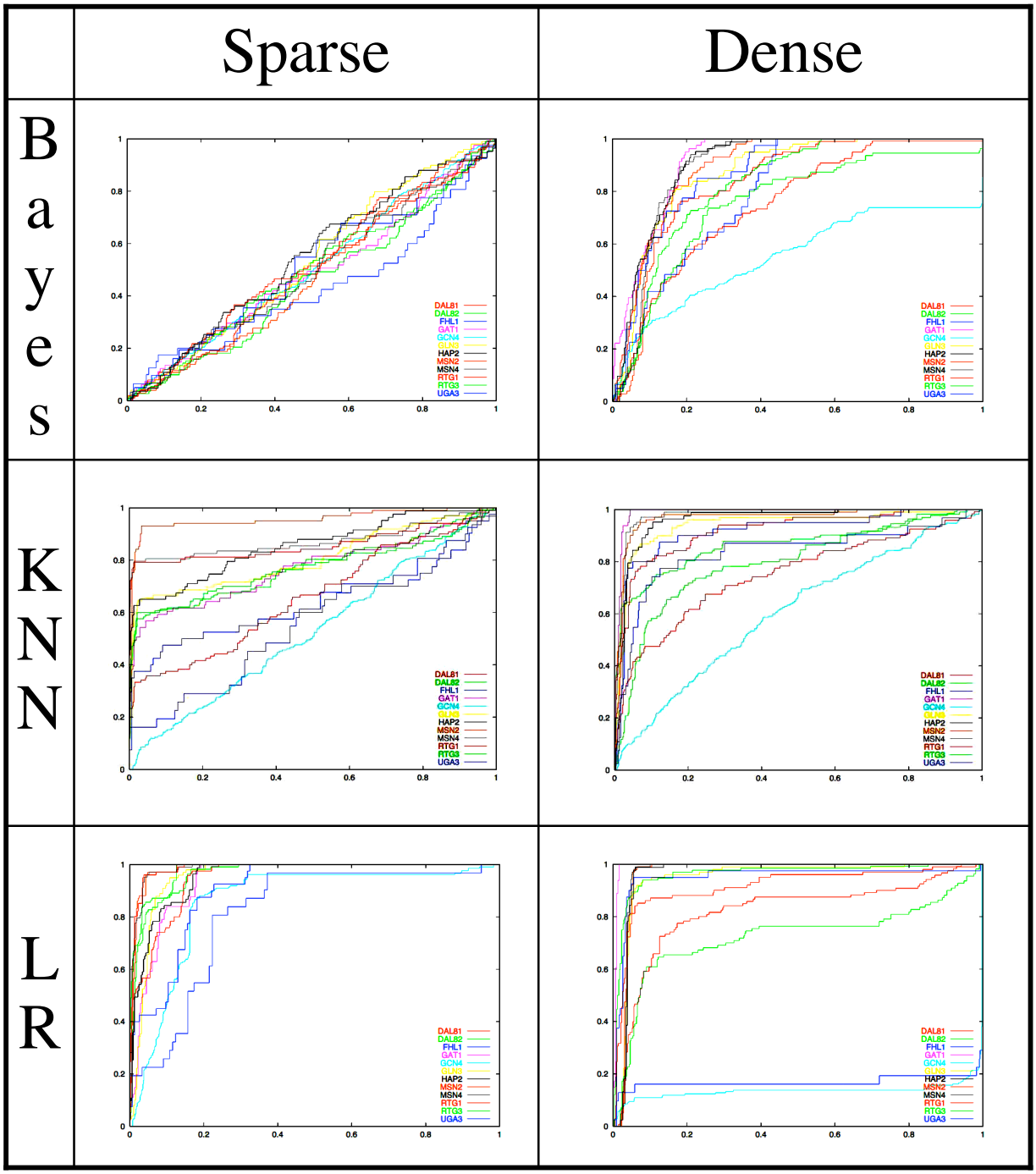

Figure 7: The ROC curves for leave-one-out experiments. These experiments indicate how well a classifier would predict the binding of a transcription factor under rapamycin conditions with only binding data for the YPD condition. The results are generally mixed with some transcription factors proving to be easier to predict than others. In particular, several classifiers have trouble with GCN4.

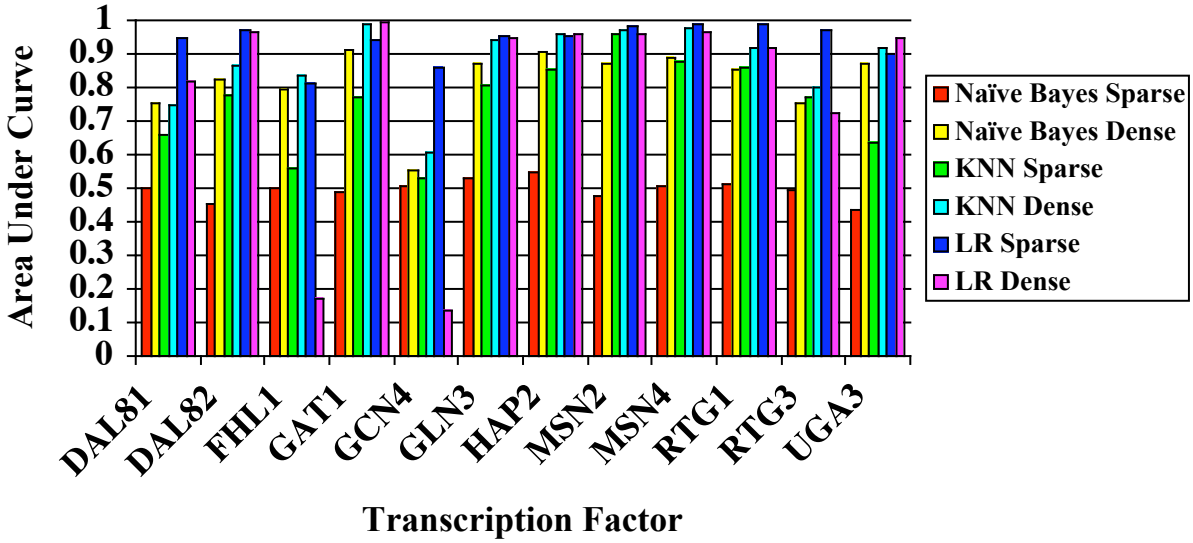

Figure 8: The area under the ROC curves of Figure 7.

classifier. The naïve Bayes with dense features performs better than random, but generally not better than the alternative classifiers. All classifiers demonstrate highly variable accuracy depending upon the transcription factor being tested. This is not surprising as different transcription factors may demonstrate different patterns of behavior. If the training data set does not contain transcription factors with similar patterns of behavior the classifier cannot predict the behavior of the new factor. This demonstrates a key limitation of this approach; the results are heavily biased by the training set. As an example, GCN4 is poorly predicted by all the classifiers. In particular, logistic regression using the dense features does worse than a random classifier when attempting to predict both GCN4 and FHL1's binding. Another severe drawback of using this approach to predict the results of an experiment is that, while the false positive *fraction* can be fairly low, the number of non-binding pairs is much larger than the number of binding pairs and so in absolute terms there will be a large amount of erroneous data.

5.3 Experiment Recommendation

The somewhat qualified success of at least some of the classifiers at predicting binding in unknown transcription factors suggests that this approach may be useful for recommending further experimentation. A researcher studying a cell's response to a previously unstudied condition might gather binding data for transcription factors which are considered likely to be involved in the cell's response to the condition, then use this data to train a classifier which would then predict which additional transcription factors are going to change in behavior the most. These factors would be the ones most likely to provide further insight on the cell's response to the new condition.

In order to evaluate the utility of using classifiers to guide experimentation, we train the logistic regression classifier using the sparse feature representation and then use all the transcription factors as a test set.[1] Ideally we would consider all possible pairs and also count those pairs which start binding in rapamycin conditions, but this isn't computationally practical. In order to keep the test set to a reasonable size we consider only those pairs of transcription factors and genes which are known to bind in YPD conditions. We then count the number of pairs which are predicted to stop binding in rapamycin conditions and rank transcription factors by the percentage of binding pairs which stop binding. We apply a simple threshold of .5 to the output of the classification data. Past results indicate this threshold is likely to get the majority of true positives while selecting few false positives.

[1]The output of the classifier is available at `http://www.cs.cmu.edu/~dkoes/research/binding/predictions`.

Transcription Factor	Change in Binding	PubMed Hits
FHL1	94%	0
GA1T	93%	7
DAL82	91%	0
UGA3	90%	0
RAP1	88%	4
MSN4	82%	3
MSN2	82%	5
ABF1	79%	1
HAP2	67%	0
CIN5	64%	0
DAL81	62%	0
GLN3	61%	21
RTG1	60%	7
REB1	59%	0
MCM1	48%	0
FKH1	47%	1
RCS1	46%	0
SWI4	44%	0
RTG3	43%	4
FZF1	41%	0

Table 1: The top 20 most differentially binding transcription factors. The percentage refers to the percent of transcription binding pairs which bind in YPD and are predicted to stop binding in rapamycin conditions. In addition, we include the results of searching for the transcription factor name and rapamycin in the PubMed database. Bold transcription factors are those factors present in the training data.

In Table 1 we list the top 20 most differentially binding transcription factors we found. Eleven of the 12 training transcription factors, which were known to play a role in the cell rapamycin response, were found. We did not identify GCN4 as a transcription factor of interest. This is not too surprising as this transcription factor proved to be difficult to predict in previous experiments.

5.4 Understanding Differential Binding

None of the classifiers build intuitive models of transcription factor binding. As such, they do not directly provide insight into why transcription factors change their binding behavior. We can, however, attempt to get an idea of which feature attributes are the most important by omitting attributes from the training data and examining the effect on the classifier. This strategy does not provide insight into the significance of the interactions between the features.

We have created modified versions of the dense feature representation training set so that each feature is omitted from the data. We've then performed 4-fold validation across each training set using the k-nearest neighbors and logistic regression classifiers. The results are shown in Table 2 and Table 3. Both classifiers found that the average expression of genes that bind to the target gene in rapamycin conditions was an important attribute for classification. Surprisingly, the remaining features did not seem to affect the result of the classification significantly with the omission of several features actually resulting in a slight improvement of classification.

Another approach to attempting to understand differential binding is to use a classifier which builds an model that is capable of providing intuition as to what is causing the changes in binding. Towards this end we learned a Bayes network from the training data, shown in Figure 9. In this model the `rap_bind` attribute has three parents, `target_ave_expr_up_YPD`, `ypd_bind`, and `target_expr_rap`, and two descendants, `target_ave_expr_up_RAP` and `factor_ave_expr_up_RAP` suggesting that knowing just these features should be sufficient to predict binding in the rapamycin condition. Unfortunately, we were unable to perform Baysian inference using this model so it remains unvalidated.

Feature Omitted	Area Under ROC Curve
target_ave_expr_up_RAP	0.674545
target_ave_expr_pp_YPD	0.912467
factor_ave_expr_pp_YPD	0.915117
factor_ave_expr_down_RAP	0.915205
factor_ave_expr_down_YPD	0.915547
ypd_bind	0.915729
target_ave_expr_down_YPD	0.915774
none	**0.915775**
target_ave_expr_down_RAP	0.915813
factor_ave_expr_up_RAP	0.916022
factor_ave_expr_up_YPD	0.916751
factor_expr_rap	0.917525
factor_expr_ypd	0.917787
factor_ave_expr_pp_RAP	0.91929
target_ave_expr_up_YPD	0.921853
target_expr_ypd	0.923719
target_ave_expr_pp_RAP	0.930527
target_expr_rap	0.941032

Table 2: The area under the ROC curve for 4-fold validation when a given feature is omitted from the training set using a k-nearest neighbor classifier on dense data. A low value suggests that the feature plays in important role in classification.

Feature Omitted	Area Under ROC Curve
target_ave_expr_up_RAP	0.604994
factor_ave_expr_pp_YPD	0.814264
factor_expr_rap	0.825819
target_expr_rap	0.826394
target_ave_expr_pp_YPD	0.829185
ypd_bind	0.829578
factor_ave_expr_up_YPD	0.829813
factor_ave_expr_up_RAP	0.829939
target_ave_expr_down_RAP	0.830015
target_expr_ypd	0.830048
factor_ave_expr_down_YPD	0.830106
factor_expr_ypd	0.830176
target_ave_expr_down_YPD	0.830221
none	**0.83024**
target_ave_expr_pp_RAP	0.830278
target_ave_expr_up_YPD	0.830424
factor_ave_expr_pp_RAP	0.830481
factor_ave_expr_down_RAP	0.830535

Table 3: The area under the ROC curve for 4-fold validation when a given feature is omitted from the training set using a logistic regression classifier on dense data. A low value suggests that the feature plays in important role in classification.

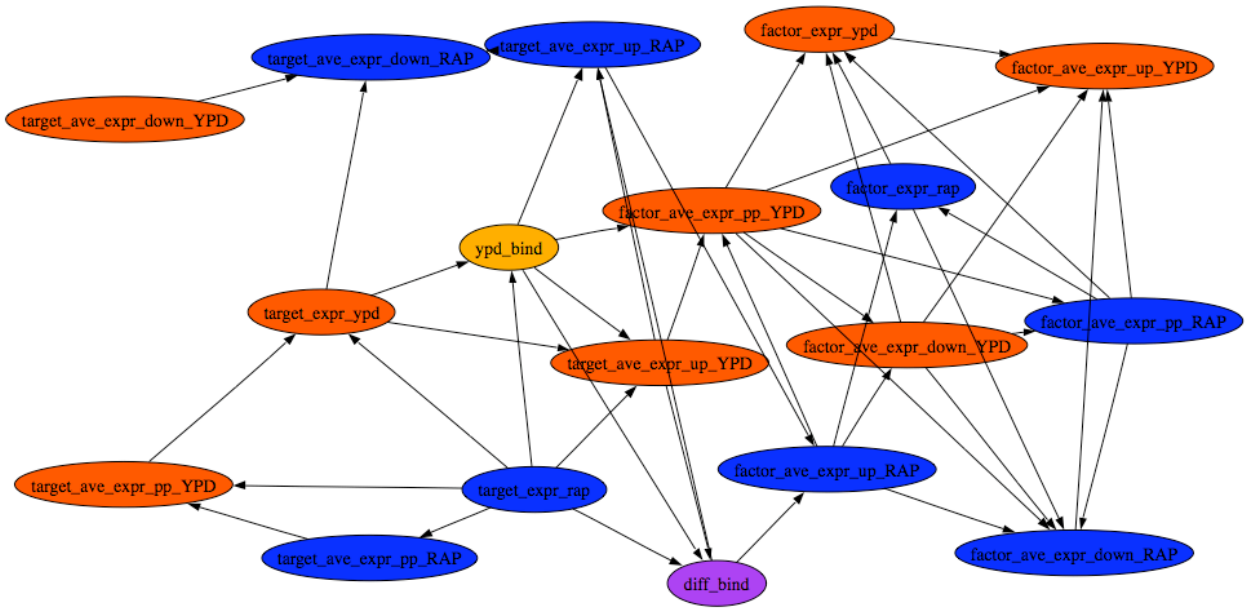

Figure 9: A Bayes network learned from the dense training data.

6 Conclusions

We find that simple classifiers, especially the logistic regression classifier using the sparse feature representation, can predict missing experimental values with high accuracy. Classifiers may also be able to predict the results of an entire experiment, but because of the number of false positives and the bias introduced by the training set they may be more suitable as a guide for further experimentation. Finally, we were not successful in extracting a deeper understanding of differences in transcription factor binding under different conditions from the classifiers.