MACHINE LEARNING FOR GENE BEHAVIOUR CLASSIFICATION

Karan Mangla Arunanshu Roy

December 12, 2008

Abstract

An interesting problem in biology is to determine the activation phase of genes. The human body contains nearly 20,000 genes. Many of these genes play a wide variety of roles in the cell cycle. Identifying the phase of activation of the gene can help determine its function in the gene. Experimentally determining the phase of all these genes can be expensive. Our tool attempts to use machine learning to determine the activation phase of a gene based on its expression profile. We first normalize the genes to be able to capture activation and deactivation. Next we apply SVM on this dataset using a small number of preclassified genes as our training set. Since genes can be active in multiple phases, we created a separate classifier for each activation phase.

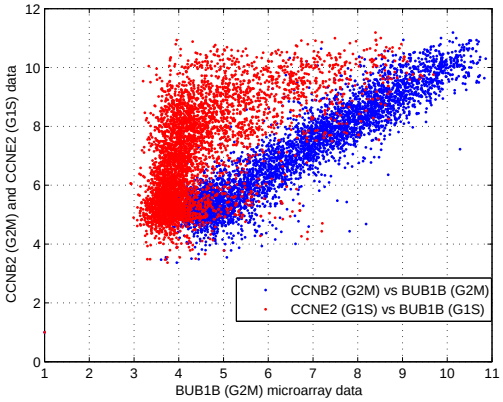

Figure 1: Plot of BUB1B (G2M) data vs CCNB2 (G2M) and CCNE2 (G1S) data

1 Introduction

The cell cycle is one of the most important gene processes, involving a large number of genes in a wide variety of functions. Gene transcription in the cell cycle occurs in phases, with each phase having its own specific transcription factors. Understanding the genes involved in each phase, would provide a better understanding of the cell cycle and also make it easier to identify the function of the gene. Running experiments to find the phase for all 20,000 genes would be quite time consuming. Our tool provides a fast machine learning approach to classify genes to cell cycle phases, based on their expression profiles in microarray experiments.

Our data consist of 4,787 Affymetrix U133Plus 2.0 human microarrays [4]. The datasets were normalized using the standard Robust Multi-chip Analysis(RMA) [3]. In total, we have 57,000 gene profiles, as genes are associated to multiple probes. We have also obtained set of 607 genes classified into cell cycle activation phases, corresponding to 1712 probes in the microarray. Genes were classified into the G1/S, G2, G2/M, M/G1 and S cell cycle phases. As can be observed in Figure 1, genes in similar phases appear to be correlated while genes in different phases are less so.

2 Errors in data

One of the major issues in tackling this problem is the inaccuracy of the data. There are various reasons for errors in the data.

- Errors in Expression Profiles: While the expression data was carefully collected there are many sources of errors in this data. Firstly, the experiments were not conducted on synchronized cultures. Hence the cultures would have contained a mixture of cells in different phases at time of collection of expression data. This will greatly reduce the strength of phase signal in the expression profiles
  Further, microarray analysis suffers from errors due to dead probes and other problems that corrupt the expression profiles.

- Errors in classified data: Our collection of classified genes are also expected to have errors due to the difficulty in experimentally classifying genes. It is estimated that 20% of our classified genes are incorrectly placed into activation phases. This will also greatly hamper the accuracy of the algorithm.

3 Experimental Procudure

We split our data randomly into 80% training data and 20% test data. All algorithms were trained and tested only on the training data. Testing of algorithms was done by creating a hold-out data set which was used to evaluate the model learned by the algorithm on the remaining training set. Our best algorithm was then verified on the 20% test data.

4 GDA

We first applied GDA to the problem. We used the simple GDA algorithm to the data. However, running the GDA algorithm over the entire data gave a covariance matrix that was nearly singular, making the calculation error-prone. Using standard PCA analysis we reduced dimensionality of the input data. After running PCA, the top k components were chosen as input for the learning algorithm.

In order to measure our results, we chose to compare precision and recall rather than accuracy. Since for each phase the proportion of genes not active in that phase are much larger than the proportion of genes active in that phase, even classifying all the genes as not active in the phase gives high accuracy, which is not actually desirable.

We observed that our system had high variance. To reduce the number of parameters in the system and prevent overfitting, we applied the Naive Bayes Assumption on our data. $P(x^{(i)}|y = 1)$ and $P(x^{(i)}|y = 0)$ were modeled as Gaussian distribution which were independent for all i. Thus the number of parameters learned is reduced and prevents overfitting.

This method was applied for a variety of number of input parameters. The results have been shown in Figure 2. We have plotted precision and recall values obtained in the different runs. We observe that the overall F-values are very low for GDA. This is expected since our features are unlikely to be independent.

5 Bayesian Logistic Regression

Since generative algorithms were performing quite poorly on our problem we attempted to apply discriminative algorithms. We applied Bayesian logistic regression using an online software [2].

We observed improved results suggesting that our data cannot be represented as a Gaussian distribution. Note that while the individual values of the precision and recall are not higher than those observed in the earlier algorithms, the F-value of our results is much better. The Table 1 provides a tradeoff between precision and recall using Bayesian Logistic Regression for the G2M phase.

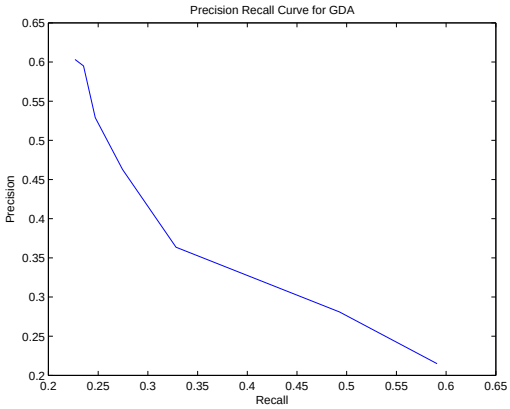

Figure 2: GDA results for classification of G2M genes

recall	precision
73.7	51.6
57.8	54.6
51.9	55.9
44.8	57.5
11.0	70.8

Table 1: Precision and Recall Tradeoff for Logistic Regression

We used Pearsons correlation to try to select good features for this dataset. However, as can be seen from from Table 2 feature reduction did not improve our results, implying that most features are useful in evaluating the activation phase of the gene.

6 Support Vector Machines

Support vector machines were used to attempt classication of the genes due to their inherent advantage in dealing with large feature spaces and large training sets. SVMs also give us the option of trying out different kernels that suit our data and has the advantage of handling outliers through the use of regularization. SVM software written

k	recall	precision
200	53.2	47.4
500	55.2	51.2
800	53.9	47.4
1000	54.50	49.4

Table 2: Precision and Recall for different number of features

Phase	Recall	Precision
G1S	61	53
MG1	46	39
G2M	58	54

Table 3: SVM based classification results on hold out set

in MATLAB [1] which is available online was used for these simulations. The first classication of G1S genes was done using a linear kernel. The results from the SVM were superior to the other supervised learning techiniques that we tried. We achieved a precision of 55% and a recall of 52%. We then used a modified kernel using the Pearson correlation between feature vectors. This improved our precision to 58% and recall to 54%.

All the 4787 features were used in these computations. However some feature reduction techniques have been tried on this data as explained in the next section and we observed that similar precision and recall can be achived with fewer features but there is no significant improvement using feature reduction. The results were not affected by value of C lying between 1 and 100 and the value of the regularization parameter is chosen to be 100. Table 3 shows the results for classication of three phases using this scheme. We get the best results for G2M and G1S phases for which we have the largest amount of classified genes.

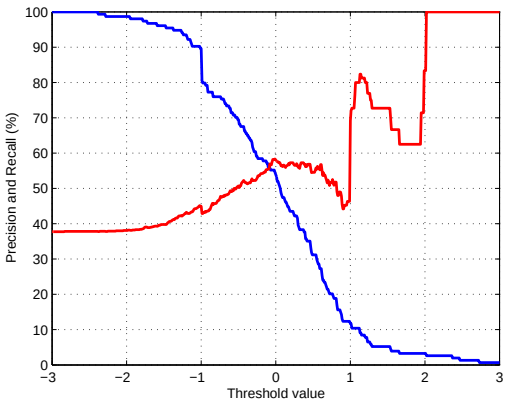

Figure 3: Precision and Recall for G2M classification using SVM

7 Feature selection

Since the best results obtained could only achieve about 60% precision and recall on the data we tried feature selection and data cleaning techniques to improve classification.. The results are for the classification of genes in G2M since the largest amount of data is available for that class.

Number of features used	Recall	Precision
100	55	45
300	51	49
500	48	47
1000	49	44

Table 4: SVD based feature reduction results on hold out set

Rank	Recall	Precision
100	48	51
100	51	59
200	46	42
500	58	54

Table 5: Low rank approximation denoising results

7.1 SVD based feature selection

Singular Value Decomposition was used to select a reduced set of features for classification. It was observed that reducing the number of features did not help to improve the classification of genes in G2M. The results are summarised in Table 4.

7.2 Low rank approximations to denoise training data

Using singular value decomposition on training data, low rank aproximations were obtained. We expect this to help reduce the noise that may be present in the training data. The results however showed that this method of denoising did not improve classification. The results are summarised in Table 5.

7.3 Removing low variance data to filter out dead probes

The microarray data was filtered to remove genes for which the data showed a low variance. We believe that this can be indicative of a dead probe and hence using a minimum threshold variance, the data was filtered to remove probes with a low variance in measurements. No significant improvement could be obtained with this technique. The results are summarised in Table 6.

8 K-Nearest Neighbors

Given that linear classifiers such as SVM and logistic regression were working very poorly on this data, we tried

Correlation threshold	Recall	Precision
0.1	56	51
0.2	52	48
0.3	46	44

Table 6: Low variance filtering results

Phase	F-Value
G1S	54.3
G2	35.9
MG1	45.5
S	37.0
G2M	57.7

Table 7: Precision and Recall for each Phase for 1-NN

non-linear classifiers in an attempt to improve the accuracy of our classification. One classifier that we considered was 1-NN. This classifier would help reduce the inaccuracy caused by classification errors in our training set as it uses more local characteristics to learn models. We ran this algorithm using the Weka software [5]. 1-NN however performed quite purely on our problem. We have provided the per phase F-values in Table 7 received for 10-fold cross validation.

9 Decision Trees

Another popular non-linear classifier is Decision trees. We tested whether using simple decision trees would allow our data to improve. Again we observed quite poor results. We tried to apply the ADA Boost algorithm to improve our decision trees. However, their was very little improvement in classification. Both these algorithms were run using Weka implementations [5].

Results obtained for ADA Boost for various boost iterations are given in Table 8. Classification was only done for the G2M phase of the cell cycle.

No. of Iterations	Precision	Recall
10	46.2	38.3
30	46	44.5
50	48.2	43

Table 8: Precision and Recall for different number of Boosting Iteration in ADA-Boost

10 Discussion on Data

Since no improvment could be obtained through the above feature selection and data denoising techniques we attempted to understand the data better through a study of correlation among data belonging to different classes. It appears that while genes belonging to the same activation phase are often well correlated, there may be a good correlation with genes belong to other activation phases too. The plots here show the correlation between the genes classified as G2M and those classified as G1S. The correlation between G2M and G1S genes is also shown. We see that several genes in G1S are well correlated with genes classified as G2M. This illustrates why the classification of genes based on the available microarray data is inherently a difficult problem. Given that the data may contain misclassified genes and other sources of noise we believe that it is difficult to achieve better classification using this data.

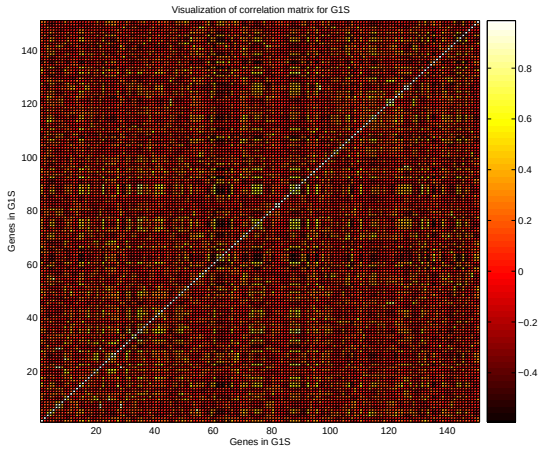

Figure 4: Correlation between G1S genes

11 Final Results

Having noted that the SVM and the Bayesian Logistic regreassion perform best in classifying our data. we ran the classification on the initially described test set. THe SVM achieved a precison of 56% and a recall of 60% while the Bayesoan logistic regression showed a precision of 57% and 59% recall.

12 Conclusion

As we can observe the data is very noisy. Hence, we get better classification results from linear classifiers than

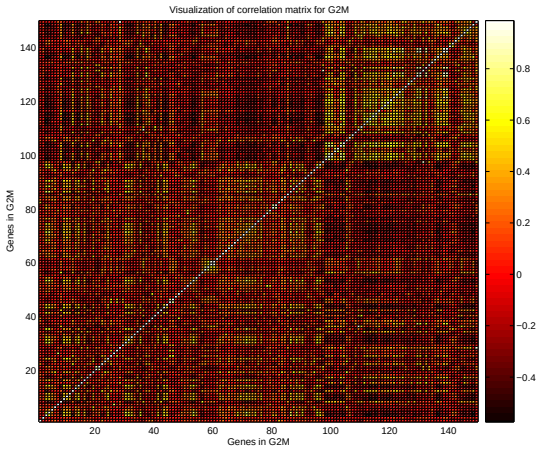

Figure 5: Correlation between G2M genes

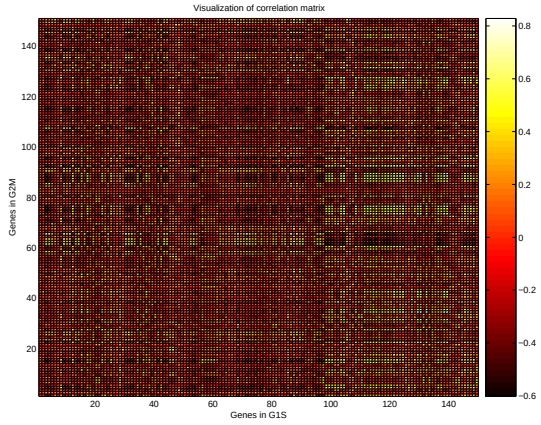

Figure 6: Correlation between G1S and G2M genes

non-linear classifiers. The best results are obtained on using SVM's on this data. Normalization of the data helps give a small increase in accuracy of the classifier. However, feature selection techniques completely fails on this dataset, suggesting that most features are important for classification. Finally, attempts to denoise the data were also not successful.

We, however, noted that results improved as the number of training examples for a phase that we could provide increased. This suggests that with a larger training set this system would provide increased accuracy. Another major problem with this data is the fact that there are errors in the classification of the training set. A more accurate training set should also boost the usefulness of this system

While there are a lot of problems with misclassification, we see that we do get good results which means that using this algorithm to prefilter genes will help to classify genes in to their correct phases. It is important to note that of the 57,000 gene probes only 1712 have been classified as yet. Thus we believe that inspite of the modest precison and recall attained, the tool can be used to classify a large number of genes and greatly increment the current information on the activation phases and roles of genes in the cell cycle.

References

[1] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France, 2005.

[2] Alexander Genkin, David D. Lewis, and David Madigan. Bbr: Bayesian logistic regression software.

[3] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of affymetrix genechip probe level data. *Nucleic Acids Res*, 31(4), February 2003.

[4] Debashis Sahoo, David L. Dill, Andrew J. Gentles, Robert Tibshirani, and Sylvia K. Plevritis. Boolean implication networks derived from large scale, whole genome microarray datasets. *Genome Biology*, 9:R157+, October 2008.

[5] Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. The waikato environment for knowledge analysis.