Using Feature Selection Filtering Methods for Binding Site Predictions

Yi Sun, Mark Robinson, Rod Adams, Rene te Boekhorst, Alistair G. Rust, Neil Davey
Science and Technology Research Institute
University of Hertfordshire, UK
{Y.2.Sun, M.Robinson, R.G.Adams, R.teBoekhorst, N.Davey}@herts.ac.uk
Arust@systemsbiology.org

Abstract

Currently the best algorithms for transcription factor binding site prediction are severely limited in accuracy. In previous work we applied classification techniques on predictions from 12 key prediction algorithms. In this paper, we investigate the classification results when 4 feature selection filtering methods are used. They are Bi-Normal Separation, correlation coefficients, F-Score and a cross entropy based algorithm. It is found that all 4 filtering methods perform equally well. Moreover, we show that the worst performing algorithms are not detrimental to the overall performance.

Keywords: *Feature Selection, Filters, Support Vector Machines, Transcription Factors.*

1. INTRODUCTION

In this paper we address the problem of feature selection for imbalanced data, in the context of improving the identification of transcription factor binding sites on sequences of DNA. There are many different algorithms to search for binding sites in current use (see Section 3). However, most of them produce a high rate of false positive predictions. This is problematic for practicing biologists who wish to validate these results - testing a prediction is costly.

In [1], we attempt to reduce these false positive predictions using classification techniques taken from the field of machine learning. We combine random selection under-sampling and SMOTE over-sampling techniques to cope with the imbalanced dataset. In addition, we use a 'window' of consecutive results in order to contextualise the neighbouring results. In this work, we investigate the classification results when using 4 feature selection filtering methods: *Bi-Normal Separation (BNS), correlation coefficients (CC), F-Score* and a cross entropy based algorithm, on the windowed inputs.

2. PROBLEM DOMAINS

It is increasingly acknowledged that the variation in complexity of organisms is due to differences in the regulation of gene activity rather than to differences in the genetic specifications for protein coding per se. Gene activity is dynamic and affected by, among other things, metabolic products and intermediates such as various hormones. Whereas the general principles underlying the translation of the coding regions of genes (exons) into their protein products are largely comprehended, the mapping between a gene's expression and the information contained in (non-coding) regulatory regions of the genome is not well understood. These regulatory regions are short sequences upstream or downstream of the position where gene transcription begins. They are generally composed of dense clusters of so-called transcription factor binding sites (TFBS). In turn, these binding sites are recognized by transcription factors, proteins that – upon binding to them - act as repressors or activators, thus controlling the rate of transcription.

Recent research has made clear that genetic regulatory mechanisms are much more intricate than was once assumed. For example, a single base substitution will commonly modify the intensity of the interaction between transcription factor and DNA rather than abolish it. This implies that such regions are fairly robust to mutations. It also allows a relatively small number of transcription factors to produce a multitude of patterns of gene expression. Furthermore, certain weakly binding transcription factors require assistance of other, more vigorously binding proteins whereas others compete for access to a single regulatory site. The situation is further complicated by the fact that certain regulatory regions are more accessible to transcription factors than others [2]. In higher eukaryotes some of these regions may be located far upstream or downstream of the target gene. These are called enhancers or cis-regulatory modules and have proven to be very difficult to recognise.

One of the most exciting, but also challenging areas of current biological research is therefore devoted to the understanding of the regulation of gene expression. The identification of regulatory regions and transcription factor binding sites clearly forms an essential step in this endeavour. However, although as much as 50% of the human genome is estimated to be regulatory [3], most of this is not yet deciphered. The desire for large scale understanding has driven the development of high throughput methods. It favours computational approaches because these sidestep the ultimately more reliable but slow and expensive route of experimental verification.

Regulatory regions appear to have statistical properties that help to distinguish them from other parts of the genome, such as the over-representation of similar sequential motifs [4-6] and a sequential persistency and an informational entropy that is intermediate between those of exons and non-coding, non-regulatory DNA [7]. These and other statistical properties are exploited by various types of algorithms for predicting TFBS, or their motifs, from raw sequence data. *Enumerative algorithms* build or assume a background model of base pair distribution in the DNA non-coding regions that do not contain TFBS, and look for motifs in the given sequence that are statistically significant against this background. They are often applied to (putative) co-regulated genes found by expression (micro-) array analysis. Another enumerative approach is *phylogenetic foot-printing*, which identifies motifs by comparing sequences from phylogenetically related species. *Iterative algorithms* use techniques such as Expectation Maximization to define weight matrices for the most over-represented motifs. These algorithms also require a collection of upstream sequences from possibly co-regulated genes and a model for background distribution. *Content based algorithms* segment the available sequence into a 'lexicon' of words and look for regularities in the way one would proceed to decipher a text consisting of a long string of letters written in an unknown language in which words are not delineated.

The downside of the developments sketched above is that we are currently burdened by a bewildering variety of algorithms. Nowadays it takes quite some computational and statistical expertise to make an educated choice about what methods to use. Even more worryingly is the fact that many of the published algorithms are still severely limited in accuracy and of uncertain quality. Not only is picking regulatory regions out of the background of other non-coding DNA sequences a non-trivial enterprise, also the fierce competition in the prediction market hardly allows for a thorough evaluation. For example, in a large sample of annotated yeast promoter sequences, a selection of 12 key algorithms were unable to reduce the error rate of positive predictions below 80%, with between 20% and 65% of annotated binding sites recovered. These algorithms represent a wide variety of approaches to the problem of transcription factor binding site prediction, such as the use of regular expression searches, PWM scanning, statistical analysis, co-regulation and evolutionary comparisons.

One way to overcome this problem is to combine the outcomes of a large number of algorithms instead of relying on the result of just one. The importance of such meta-classifiers goes without question and their investigation will therefore be at the core of this paper. In the work described here we take the results from the 12 aforementioned algorithms and combine them into 2 different feature vectors, as shown in next section. We then investigate whether the integrated classification results of the algorithms can produce better classifications than any one algorithm alone. (See Figure 1, and more details about Figure 1 can be found in section 3.1). In our previous work [1], we found that the integrated classifier using a *support vector machine* (SVM) [8] outperform each of the original individual algorithms and the other classifiers employed in this work. In particular they have a better tradeoff between recall and precision.

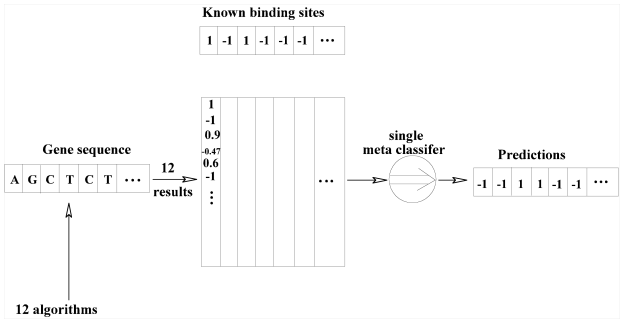


Figure 1. The 12 algorithms give their own prediction for each sequence position and one such column is shown. The 12 results are combined as an input to a classifier.

3. DATASETS AND METHOD PERFORMATIONS

3.1 Description of the Data

The data is extracted from the *SCPD* database (http://cgsogma.cshl.org/jian), which is one of the largest

and most reliable collections of experimentally verified annotated data available. The dataset has 68910 possible binding positions and a prediction result for each of the 12 base algorithms. The 12 algorithms can be categorised into higher order groups: *Single sequence* algorithms (7) [9-12]; *Coregulatory* algorithms (3) [13, 14]; A *Comparative* algorithm (1) [15]; An *Evolutionary* algorithm (1) [16]. The data has two classes labeled as either binding sites or non-binding sites, with about 93% being non-binding sites.

In this work, we use 2/3 of the data as the training set and 1/3 as the test set. Amongst the data there are repeated vectors, some with the same label (repeated items) and some with different labels (inconsistent items). It is obviously unhelpful to have these repeated or inconsistent items in the training set, so they are removed. However there is no change in the case of the test set, which therefore contains the full set of data.

As the data is drawn from a sequence of DNA nucleotides the label of other near locations is relevant to the label of a particular location. We therefore contextualise the training and test data by windowing the vectors as shown in Figure 2. We use the locations up to three either side, giving a window size of 7, and a consequent input vector size of 84. This has the considerable additional benefit of eliminating most of the repeated and inconsistent data.

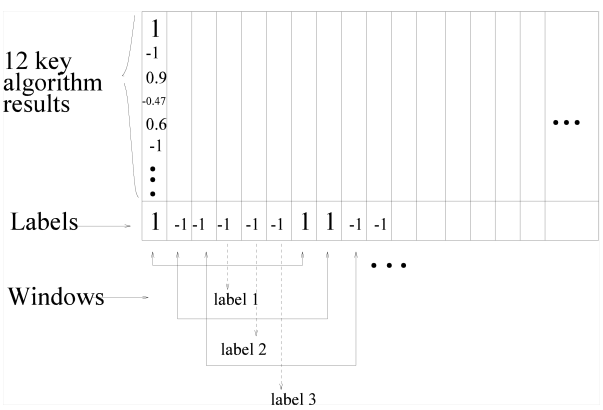


Figure 2. The window size is set to 7 in this study. The middle label of 7 continuous prediction sites is the label for a new windowed inputs. The length of each windowed input now is 12 × 7.

3.2 Sampling Techniques for Imbalanced Dataset Learning

Since the dataset is *imbalanced*, supervised classification algorithms will be expected to over predict the majority class, namely the non-binding site category. There are various methods of dealing with imbalanced data [17], classified as algorithm-based and data-based methods. So far we concentrate on data-based methods. In this work we apply random selection under-sampling for the majority class (negative examples) and SMOTE [18] over-sampling for the minority class (positive examples). More details on data-based methods can be found in [19].

The aim of the SMOTE method is to synthetise new patterns by applying majority voting to each of the attributes of the K-nearest neighbours of each pattern in the minority class. We take 5 nearest neighbours, and triple the number of items in the minority class. The actual ratio of minority to majority class is determined by under-sampling the majority class. For feature selection to work effectively it is desirable for the two classes to be of equal size.

3.3 Classifier Performance

To evaluate classifiers used in this work, we apply a range of standard reference metrics defined in Table 1, where *TN* is the number of true negative samples; *FP* is false positive samples; *FN* is false negative samples; *TP* is true positive samples.

Table 1. Definitions of several performance metrics.

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - Score = \frac{2 \cdot Recall \cdot Precision}{Recall + Precison}$$

$$FP\_Rate = \frac{FP}{FP + TN}$$

$$CC = \frac{TP \cdot TN - FN \cdot FP}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}}$$

In the context of identifying binding sites a high *Precision* and low *FP_Rate* are particularly important, as a higher cost is associated with a degradation of performance on these metrics. There is a trade-off between Precision and *Recall* for the imbalanced dataset, integration of the metrics using the *F-Score* provides a single metric for evaluating overall performance. The value of the *correlation coefficient*, *CC* [20], ranges from -1 to 1. When predictions exactly coincide with the known binding site, it has value 1.

4. FEATURE SELECTION

It is known that a dataset with a large number of features may suffer from the curse of dimensionality

[21]. To alleviate this problem, many feature selection techniques have been proposed. One excellent recent introduction to this topic can be found in [22]. There are three main approaches: *wrappers*, *filters* and *embedded* methods. Wrappers and embedded methods integrate feature selection with the machine learning process, while filters are a pre-processing step, which choose a subset of features according to a particular feature selection algorithm. In this work, we focus on filtering methods.

In the context of the data used here, feature selection is the elimination of one or more of the base algorithms that may be less useful in constructing the final meta-classifier. This elimination can be achieved simply by using one (or more) of the aforementioned performance metrics. In this way we are selecting the best algorithms prior to combining them to produce the meta-classifier. Two suitable metrics are F-Score and CC as defined in Table 1.

Alternately, we can select features to eliminate by using a combination of the metrics, as described in Section 4.1, or an algorithm from Information Theory, as described in Section 4.2.

4.1 Filtering Metrics

Bi-Normal Separation (*BNS*) is a method that combines two metrics to compute which features to eliminate. BNS was proposed in [23], where it was found to perform well. Its definition is given by $\left|F^{-1}(Recall) - F^{-1}(FP\_Rate)\right|$, where F^{-1} is the standard *normal inverse cumulative distribution*. The BNS distance metric is proportional to the area under the *ROC curve* [24], which is often used to measure a classifier's performance.

4.2 The Entropy Based Algorithm

An algorithm, using cross entropy was proposed in [25]. Here we describe it following [17].

Assuming there is a set of features $V = \{V_1, V_2, \ldots, V_p\}$.

1) For each pair of features (V_i and V_j) compute the cross-entropy of the class distribution:

$$d_{ij} = D\left(\Pr(C \mid V_i = v_i, V_j = v_j), \left(\Pr(C \mid V_i = v_i)\right)\right)$$

where $D(p,q) = \sum_{x \in \Omega} p(x) \log \frac{p(x)}{q(x)}$ denotes the cross-entropy of distributions p and q, and C denotes the possible classes.

2) Iterate a)-c) until some pre-specified number of features has been deleted:

a) Let $G = V$. For each feature $V_i \in G$, choose M_i to be a set of features V_j in $G - \{V_i\}$ for which d_{ij} is smallest.

b) Calculate the expected cross-entropy for each i as follows:

$$\delta_G(V_i | M_i) = \sum_{v_{M_i}, v_i} \left[\begin{array}{l} \Pr(M_i = v_{M_i}, V_i = v_i) \times \\ D\left(\Pr(C \mid M = v_M, V_i = v_i) \Pr(C \mid M = v_M)\right) \end{array} \right]$$

c) Select the i whose quantity computed from step b) is the smallest, and define $G = G - \{V_i\}$

5. EXPERIMENTAL RESULTS

5.1 Results on Feature Selection

We compute the BNS, F-Score and CC scores for each of 12 base algorithms. The cross entropy-based algorithm is implemented using the *classification toolbox*, which is available at the URL http://www.yom-tov.info/toolbox.html. These scores are sorted and plotted in Figures 3-6.

To select a subset of the features we are looking for a suitable boundary in the graphs where the curves begin to fall away relatively rapidly. We decided that 6 features represented a good compromise. Thus, the length of a windowed feature vector is 6 × 7 = 42.

Table 2 shows that the rank of the 12 base algorithms (denoted A to L) resulting from the different feature selection methods. It can be seen that there are 4: A, C, G and I, in common in the first 6 selected features of all the filtering methods employed here. In addition, the ranking of features using the BNS and CC methods are the same except for rank 9 and 10.

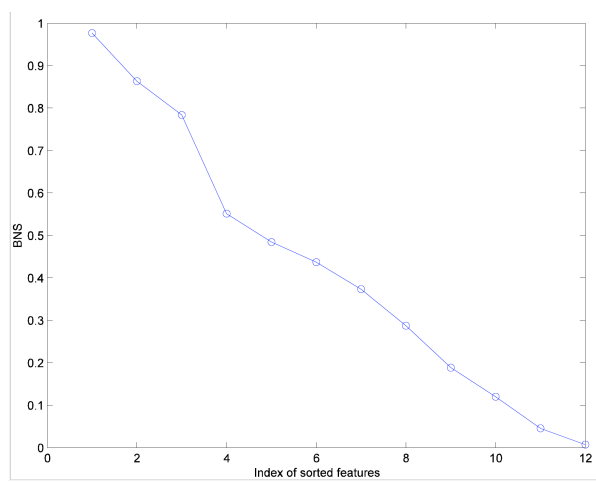
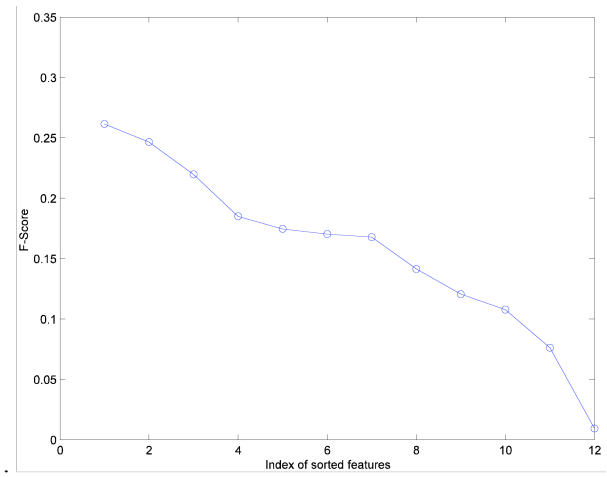


Figure 3. BNS of each algorithm.

5.2 Results on Classification

Since BNS and CC selected the same 6 algorithms there are only 3 different sets of data, one for each of the different sets of selected features. For each set of data, an SVM was trained as a meta-classifier. All the SVM parameters were obtained using cross-validation. For the purpose of comparison we also ran the SVM on the full set of 12 features, denoted by *Full* in Table 3.

Table 3 shows the results. Interestingly, we can see that all performances are similar. Full windowed inputs have the highest F-Score, while BNC/CC windowed inputs have the lowest FP-Rate and highest Precision. Overall, none of the 4 feature selection filtering methods outperforms the others.



. **Figure 4.** F-Score of each algorithm.

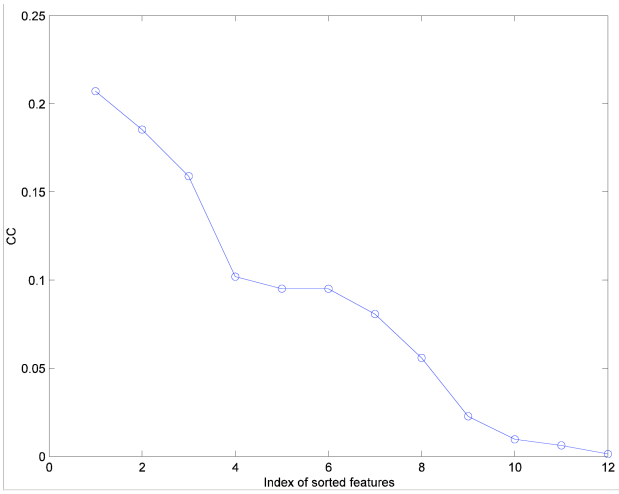
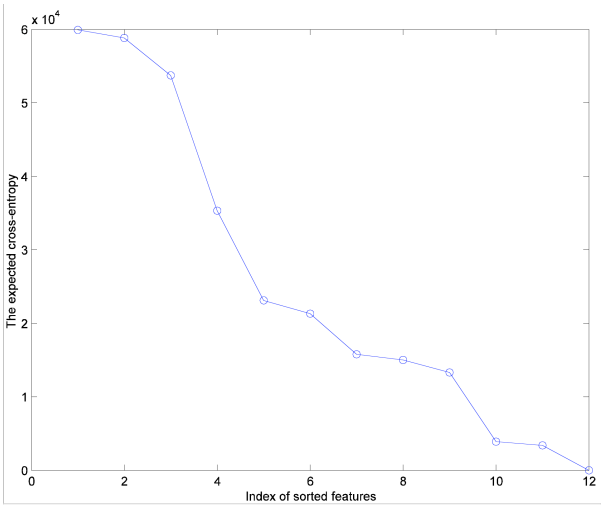


Figure 5. CC of each algorithm.



. **Figure 6.** The expected cross entropy of each algorithm.

Table 2. The rank of the 12 base algorithms.

Rank	BNS	F-Score	CC	Entropy
1	I	I	I	B
2	A	A	A	G
3	G	G	G	A
4	J	K	J	C
5	C	L	C	J
6	K	C	K	I
7	L	J	L	D
8	D	D	D	F
9	B	F	F	E
10	F	E	B	H
11	H	H	H	K
12	E	B	E	L

6. DISCUSSION

It is found that all 4 feature selection filtering methods perform well. All of them give similar classification performances to the classifier which used the full set of features. This is an interesting result since it implies, for this dataset, that the filtering process is unnecessary. It is clear that the 6 worst performing algorithms were not detrimental to the overall performance of the meta-classifier. On the other hand, we have shown that effective classification can be achieved using just the 6 best algorithms.

Table 3. Performance of the 4 feature selection methods. Each row corresponds to one feature selection method, with the performance of the resulting classifier down each column.

	Recall	Precision	F-Score	FP_Rate	CC
Full	0.35	0.26	0.30	0.07	0.24
BNS/CC	0.28	0.30	0.29	0.05	0.23
F-Score	0.29	0.27	0.28	0.06	0.22
Entropy	0.31	0.27	0.29	0.06	0.23

References

[1] Y. Sun, M. Robinson, R. Adams, P. Kaye, A. G. Rust, and N. Davey, ``Using real-valued meta classifiers to integrate binding site predictions," *Proceedings of International Joint Conference on Neural Networks,* 2005.

[2] R.J. White, *Gene Transcription: Mechanisms and Control*, Blackwell, 2001.

[3] M. Markstein, A. Stathopoulos, V. Markstein, P. Markstein, N. Harafuji, D. Keys, B. Lee, P. Richardson, D. Rokshar, and M. Levine, ``Decoding Noncoding Regulatory DNAs in Metazoan Genomes,'' *Proceeding of 1st IEEE ComputerSociety Bioinformatics Conference (CSB 2002)*, Stanford, CA, USA, 14-16, August, 2002.

[4] P. Bucher, ``Weight matrix descriptions of four eukaryotic RNA polymerase II promotor elements derived from 502 unrelated promotor sequences," *J. Mol, Biol*. 212: 563-578.

[5] T.G. Wolfsberg, A.E. Gabrieliam, A.E. Campbell, M.J. Cho, R.J. Spouge, and D. Landsman, ``Candidatge regulatory sequence elements for cell cycle - dependent transcription in Sacharomycescerevisiae," *Genome Research*, 9, pp775-792.

[6] I. Abnizova, R. te Boekhorst, C. Walter, and W. Gilks, ``Some statistical properties of regulatory DNA sequences and their use in predicting regulatory regions in Drosophil genome: the `Fluffy Tail Test'," cccepted for publication in *BMC Bioinformatics*, 2005.

[7] R.te Boekhorst, I. Abnizova, and C.L. Nehaniv, ``An adaptive sliding window algorithm for inferring DNA functionality from sequence information," submitted to *Applied Bioinformatics*, 2004.

[8] B. Scholkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press, 2002.

[9] http://www.hgmp.mrc.ac.uk/Software/EMBOSS/.

[10] A. Apostolico, M.E. Bock, S. Lonardi and X. Xu, ``Efficient Detection of Unusual Words," *Journal of Computational Biology*, Vol.7, No.1/2, 2000.

[11] N. Rajewsky, M. Vergassola, U. Gaul and E. D. Siggia, ``Computational detection of genomic cis regulatory modules, applied to body patterning in the early Drosophila embryo," *BMC Bioinformatics*, 3:30, 2002.

[12] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. De Moor, RouzP and Y. Moreau, ``A Gibbs Sampling method to detect over-represented motifs in upstream regions of coexpressed genes," *Proceedings Recomb'2001*, 305-312, 2001.

[13] T. L. Bailey and C. Elkan, ``Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 28-36, AAAI Press, 1994.

[14] J. D. Hughes, P. W. Estep, S. Tavazoie and G. M. Church, ``Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae," *Journal of Molecular Biology*, Mar. 10; 296(5):1205-1214, 2000.

[15] http://family.caltech.edu/SeqComp/index.html.

[16] M. Blanchette and M. Tompa, `` FootPrinter: a program designed for phylogenetic footprinting," *Nucleic Acids Research*, Vol. 31, No. 13, 3840-3842, 2003.

[17] G. Wu, and E.Y. Chang, ``Class-boundary alignment for imbalanced dataset learning,'' *Workshop on learning from imbalanced datasets, II, ICML*. Washington DC, 2003.

[18] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, ``SMOTE: Synthetic minority over-sampling Technique," *Journal of Artificial Intelligence Research*. Vol. 16, pp.321-357, 2002.

[19] Y. Sun, M. Robinson, R. Adams, R. teBoekhorst, A.G. Rust, and N. Davey, ``Using sampling methods to improve binding site predictions,'' accepted by *The 14 th European Symposium on Artificial Neural Network (ESANN),* 2006.

[20] M. Burset, and R. Guigo, ``Evaluaton of gene structure prediction programs,'' *Genomics* 34, pp353-367,1996.

[21] C.M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[22] I. Guyon, and A. Elisseeff, ``An introduction to variable and feature selection, '' *Journal of Machine Learning Research,* Vol. 3, pp1157-1182, 2003.

[23] G. Forman, ``An extensive empirical study of feature selection metrics for text classification,'' *Journal of Machine Learning Research*, Vol. 3, pp1157-1182, 2003.

[24] R. Fawcett, ``ROC graphs: notes and practical considerations for researchers," Kluwer Academic publishers, 2004.

[25] D. Koller, and M. Sahami, ``Toward optimal feature selection,'' Proceedings of the Thirteenth International Conference, In Lorenza Saitta, ed., Machine Learning, Morgan Kanfmann, 1996.