Using Real-valued Meta Classifiers to Integrate and Contextualize Binding Site Predictions

Offer Sharabi, Yi Sun, Mark Robinson, Rod Adams, Rene te Boekhorst, Alistair G. Rust, Neil Davey

University of Hertfordshire, College Lane, Hatfield, Hertfordshire AL10 9AB
{o.Sharabi, 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 predictions are severely limited in accuracy. However, a non-linear combination of these algorithms could improve the quality of predictions. A support-vector machine was applied to combine the predictions of 12 key real valued algorithms. The data was divided into a training set and a test set, of which two were constructed: filtered and unfiltered. In addition, a different "window" of consecutive results was used in the input vector in order to contextualize the neighbouring results. Finally, classification results were improved with the aid of under and over sampling techniques. Our major finding is that we can reduce the False-Positive rate significantly. We also found that the bigger the window, the higher the F-score, but the more likely it is to make a false positive prediction, with the best trade-off being a window size of about 7.

1 Introduction

In this paper, we investigate the effect of contextualizing data, within the framework of improving the identification of transcription factor binding sites on sequences of DNA using a Support Vector Machine (SVM). There are several algorithms to search for binding sites in current use [1]. However, most of them are severely limited in their accuracy and yield many false positive results. That imposes a serious problem for practicing biologists, as experimentally validating a prediction is costly.

In [2] we attempted to reduce these false positive predictions using classifications techniques employed in the field of machine learning. Since the data is exceptionally skewed (about 93 percent are in one class, not part of a binding site), we further dealt with the problem of classification in an imbalanced data set in [3]. Although we contextualized the data in previous work [2], the window size was fixed at 7. In this paper we extend this anlaysis for different sizes of windows. The change in outcome is reflected by a variety of performance metrics.

2 Problem Domain

There is currently a considerable research focus towards gaining a functional understanding of genomic regulatory control. Many important biological systems are

controlled, to some extent, by Gene Regulatory Networks (GRN's) and an increased understanding of their regulation and encoding would be invaluable. Transcriptional control of gene regulation is a fundamental feature of GRN's, especially so in developmental GRN's. A crucial step for increasing our understanding of processes at this level is to be able to predict the short sequences of DNA that bind transcription factors (TFBS). These sequences effectively determine the set of proteins which are able to influence the expression of a particular gene. The computational prediction of TFBS is a necessary precursor for a genome wide analysis of GRN's.

The development of algorithms for the prediction of TFBS is a difficult problem. The rules that determine the specific set of sequences which a transcription factor will bind strongly with are both non-trivial and still not fully understood. In spite of considerable improvements in the accuracy of such algorithms in recent years, the high number of false positive predictions still severely limits the utility of such algorithms. Working on the premise that algorithms with differing algorithmic foundations may well be, to some degree, complementary, we have, in previous work, explored the use of machine learning methods for integrating 12 prediction algorithms [2]. In this work we explore the importance of data contextualization by using a "window" of neighbouring predictions as an input vector (see Sections 3 and 5). In particular we explore the importance, if any, of the size of the window for improving prediction accuracy.

3 Description of the Data

The data is a sequence of 68910 nucleotides, each of which may be part of a binding site. For each nucleotide there is a prediction result from each of the 12 base algorithms, which may be either real valued or binary. Each nucleotide also has a label denoting whether it is part of a known binding site. The data therefore consists of 68910 12-ary real vectors, each with an associated label.

The data set was divided into a training set that consisted of 2/3 of the data, the remaining 1/3 was used as the test set. Amongst the data, there are repeated vectors, some with the same label (repeated items), and some with contradictory labels (inconsistent items). These items are unhelpful in the training set and were therefore removed. The filtered training set is called the consistent training set. However, in the case of the test set, both the full set of data and the subset of consistent test items are considered. The full data set was called the unfiltered data set, whereas the subset of consisted test items was called the filtered data set.

In the dataset, there are fewer than 10% binding positions amongst all the vectors, so this is an imbalanced dataset [4]. An imbalanced dataset imposes a problem for supervised classification algorithms, as they are expected to over-predict the majority class, namely the non binding site category. One of the techniques to overcome this problem is to apply the data based method: under-sampling of the majority class and over sampling of the minority class. For under sampling, a subset of data points from the majority class is randomly selected. For over sampling, the SMOTE algorithm is

used [5]. The process of integrating, sampling and classifying the data, is illustrated in Figure 1.

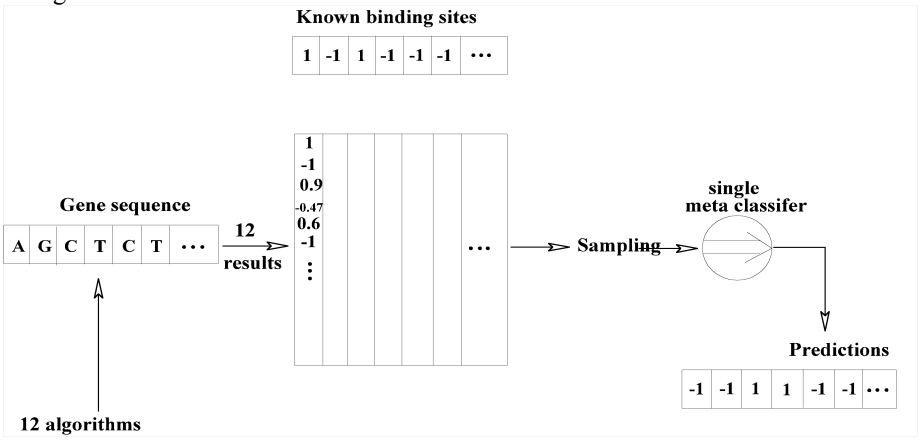

Fig. 1. The integration, sampling and classification of the data. For each location in the sequence, the prediction results of 12 the algorithms was integrated into one single vector. The data was under and over sampled, and then classified using a meta-classifier

4 Contextualizing the 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. In other words if a specific nucleotide is part of a binding site then it is highly likely that its neighbours will also be part of the same binding site. Therefore, adding the neighbouring vectors of a particular vector, windowing the vectors, can improve predictions. In [2] we used the location of the 3 nearest sites to either side of a given site, thereby constructing a window size of 7, and a consequent vector of 84 (12 times 7), as shown in Figure 2.

In this work, we used the location of K nearest locations to either side, where K = (1, 2, 3, 4, 5, 6). The result is a window size of $2K+1$, and a consequent vector of 12 times ($2K$+1). Windowing the vectors has the additional benefit of eliminating most of the repeated and inconsistent data. In bigger windows, not only is the training set and the test set sizes increased, but also the difference between both test sets decreased, as can be seen in Table 1.

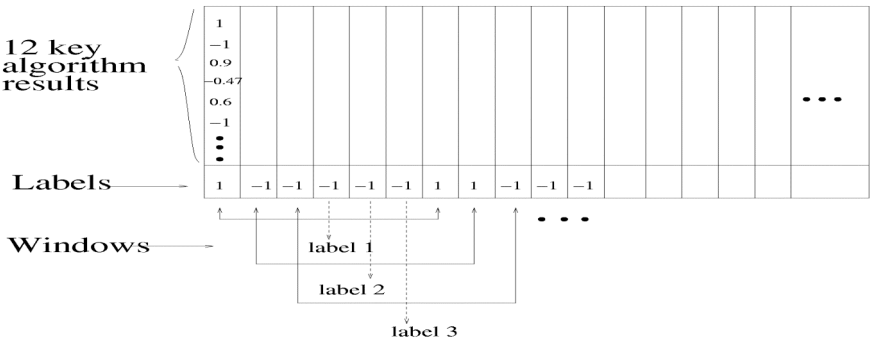

Fig. 2. Contextualising the data. In this example the window size is set to 7. The middle label prediction is the label for the windowed input. The length of each windowed input is now 12 × (2*K*+1)

Table 1. The sizes of *training-set*, *testing set* and *filtered testing set* used in this experiment. Note that as window size increases, the difference between the unfiltered and filtered test sets decreases.

Window Size	Set		
	Training	Unfiltered testing	Filtered testing
3	17701	22511	9767
5	26770	22509	14399
7	32595	22507	17233
9	36670	22505	19093
11	39503	22503	20277
13	41400	22501	21064

5 Classifier Performance

Classification accuracy rate is not sufficient as a standard measure for a problem domain with an imbalanced data. Therefore, several common performance metrics were applied, such as *Recall* (1), *Precision* (2), *FP-rate* (3), and an *F-score* (4) [6], [7]. Also, a Receiver Operating Characteristics (*ROC*) analysis [8] was applied.

5.1 Performance Metrics:

Based on the confusion matrix computed from the test results, several common performance metrics can be defined, where *TN* is the number of True Negative

samples; *FP* is the False Positive samples; *FN* is False Negative samples; *TP* is True Positive samples.

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$FP\text{-}rate = \frac{FP}{FP + TN} \quad (3)$$

$$F\text{-}score = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

Note that for all the measures except *FP-Rate* a high value is desirable. Most of the base algorithms have a high *Recall* by simply over predicting the binding site class (predicting every item to be positive gives a recall of 1), and this is problematic. On the other hand *Precision* is the proportion of the positively categorised samples that are actually part of a binding site. Increasing the *Precision* of the prediction is one of the main goals of our meta-classifier. However increasing *Precision* is normally accompanied by a decrease in the *Recall*, so the *F-score*, which takes into account both *Recall* and *Precision*, is a useful measure of overall performance. The *FP-rate* is the proportion of all the negative samples that are incorrectly predicted. The base algorithms generally have a high *FP-rate* and reducing this is another major goal of our classifier.

5.2 ROC Curves

In a ROC diagram, the true positive rate (*Recall*) is plotted on the Y-axis and the false positive rate (*FP-rate*), is plotted on the X- axis. Points in the top left corner of the diagram have a high *Recall* and a low *FP-rate* and so represent good classifiers. Often, to measure a classifier performance it is convenient to use a single metric and the area under an *ROC* curve (*AUC*) can be used for this purpose. The *AUC* value ranges between 0 to 1, where an effective classifier should have an *AUC* which is greater than 0.5.

6 Experiments

The classification technique used in this work is a Support Vector Machine [9], and the experiments were completed using LIBSVM[1] . The RBF kernel was used. The SVM therefore has two parameters, C (the penalty parameter) and γ (width of the kernel). The C values were (5, 20, 50, 100, 300, 1000, 2000) and the γ values were (0.01 0.04 0.01 0.001). The window sizes ranged from 1 to 13, in increments of 2. For each window size, the performance matrix and ROC curves of the 6 C values and of the 4 γ values were computed, both for the filtered test set and for the unfiltered test

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm.

set. For example, for window size 5, there were altogether 48 results: 24 (6 times 4) results for the filtered set, and 24 results for the unfiltered set.

7 Results:

Tables 2a and 2b contain the best results for each window size. The best results were gained when the γ value was fixed to 0.001 for all windows. The *C* value differed between the window sizes, but ranged from 5 to 300. In fact, in [10] we showed that all 4 best results for each window size were in that range, with the exception of window size 3. The results clearly show that the *FP-rate* decreased dramatically from the best algorithm to the single vector (window size 1), and decreased further when "windowing" the data. The lowest *FP-rate* was 0.005 for window size 3. Vectoring the data is useful for increased *Precision* as well. The *Precision* of window size 1 is 0.377, and higher than that of the best algorithm (0.222). The *Precision* of window-size 3 is the highest (0.649). From window size 3 onward however, the precision value slightly decreased as the window size increased. Thus, window size 3 has the lowest false positive rate, with the highest *Precision*. The AUC values of all results were higher than 0.5, meaning that the classifier preformed better than chance level. Another way to determine a good result is by comparing a high F-score to a low FP-rate. In accordance with previous experiments, [1], [2], [3], there is a trade off between *F-score* and an *FP-rate*. When the *F-score* rises, so does the false positive rate. Bigger window sizes generate higher *F-scores* but also higher *FP-rates*. However, from Figure 3, window size 7 seems to have a good trade-off between the two measures. Furthermore, it became apparent that for the window sizes 3, 5, and 7 there is a better trade off between an *F-score* and *FP-rate* in the unfiltered set compared to the filtered set. Another difference between the test sets is that for each window the filtered test set had a little higher AUC scores than the unfiltered test set.

Table 2a. Common performance metrics (%) calculated from confusion matrices on the *unfiltered test set.* The selected best parameters for each window size are also shown in the table.

Window Size	*C*	γ	Recall	Precision	F-score	FP-rate	AUC
	Unfiltered data						
Best Alg.	-	-	0.400	0.222	0.285	0.106	-
1	**300**	**0.001**	0.246	0.377	0.297	0.031	0.710
3	**5**	**0.001**	0.111	0.649	0.190	0.005	0.650
5	**100**	**0.001**	0.179	0.504	0.260	0.013	0.640
7	**50**	**0.001**	0.197	0.489	0.280	0.015	0.650
9	**50**	**0.001**	0.226	0.446	0.300	0.021	0.650
11	**20**	**0.001**	0.231	0.443	0.300	0.022	0.660
13	**5**	**0.001**	0.232	0.430	0.300	0.023	0.680

Table 2b. Common performance metrics (%) calculated from confusion matrices on the *filtered test set.* The selected best parameters for each window size are also shown in the table,

Window Size	*C*	*γ*	Recall	Precision	F-score	FP-rate	AUC
	Filtered data						
1	**300**	**0.001**	0.341	0.344	0.342	0.073	0.723
3	**5**	**0.001**	0.132	0.628	0.218	0.008	0.695
5	**100**	**0.001**	0.207	0.511	0.295	0.017	0.675
7	**50**	**0.001**	0.221	0.499	0.307	0.019	0.672
9	**50**	**0.001**	0.245	0.449	0.317	0.024	0.661
11	**20**	**0.001**	0.245	0.444	0.316	0.024	0.668
13	**5**	**0.001**	0.244	0.432	0.312	0.025	0.689

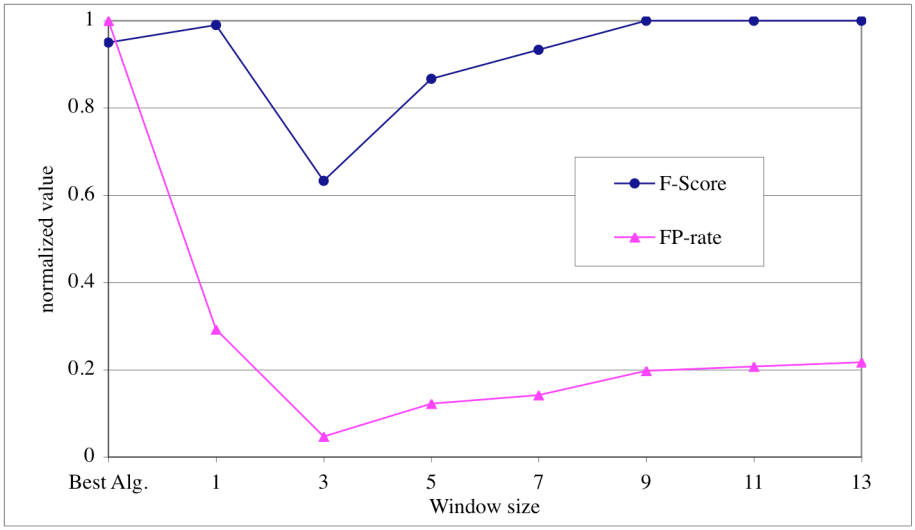

Fig. 3. Normalized (to have maximum of 1) values for *F-score* (dots) and *FP-rate* (triangles) for the various unfiltered window sizes. From window size 3 both *FP-rate* and *F-score* are increasing, as the window size increases.

8 Conclusions:

An important finding of this study is that by vectoring and later "windowing" the data, the *FP-rate* is significantly decreased, from as much as 10% to as little as 0.5%. That has important implications for experimental biologists for whom the high *FP-rates* considerably reduce the utility of these algorithms. All window sizes have a better *Precision* than that of the best base algorithm, and more importantly, a better

trade off between an *F-score* and an *FP-rate*. Window sizes affect the performance of the SVM classifier. The bigger the window size, the higher the *F-score*. However, that comes with a cost; the *FP-rate* is increased accordingly. Arguably, the best trade-off was gained for window size 7. The best values for the SVM parameters were fixed with the γ value on 0.001, but ranged for the *C* value from 5 to 300. For window sizes 3 to 7 the unfiltered data set had a better trade-off between *F-score* and *FP-rate*. However, as the window size got bigger, the difference between the test sets' sizes decreased. Consequently, the similarity between the various performance metrics increased.

References

1. Robinson, M., Sun, Y., teBoekhorst, R., Kaye, P., Adams, R. & Davey, N.: "Improving computational predictions of cis-regulatory binding sites," *Biocomputing - Proceedings of the Pacific Symposium,* (2006)
2. Sun, Y., Robinson, M., Adams, R., Kaye, P., Rust, A. G. & Davey, N.: "Using real-valued meta classifiers to integrate binding site predictions," *Proceedings of International Joint Conference on Neural Networks,* (2005)
3. Sun, Y., Robinson, M., Adams, R., teBoekhorst, R, Rust, A. G. & Davey, N.: "Using Sampling methods to improve binding site predictions." accepted by *The 14th European Symposium on Artificial Neural Network (ESANN), (*2006)
4. Japkowicz, N: Class imbalances: Are we focusing on the right issue? Workshop on learning from imbalanced datasets, II, ICML, Washington DC. (2003)
5. Chawla, N. V., Bowyer, K. W., Kegelmeyer L. O. & Kegelmeyer, W. P.: "SMOTE: Synthetic minority over-sampling Technique," *Journal of Artificial Intelligence Research.* Vol. 16, (2002) 321-357
6. Buckland, M. & Gey, F.: The relationship between Recall and Precision. Journal of the American Society for Information Science: Vol. 45, No. 1, (1994) 12--19
7. Joshi, M., Kumar, V. & Agarwal, R.: Evaluating Boosting algorithms to classify rare classes: Comparison and improvements. First IEEE International Conference on Data Mining, San Jose, CA (2001)
8. Fawcett, R.: ``ROC graphs: notes and practical considerations for researchers," Kluwer Academic publishers, (2004)
9. Scholkopf, B & Smola, A. J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, The MIT Press (2002)
10. Sharabi O.: "Using real-valued meta-classifiers to integrate and contextualize binding site predictions," University of Hertfordshire, STRC, technical report, in press.