Improving Computational Predictions of Cis- Regulatory Binding Sites

Mark Robinson, Yi Sun, Rene Te Boekhorst, Paul Kaye, Rod Adams, Neil Davey, and Alistair G. Rust

Pacific Symposium on Biocomputing 11:391-402(2006)

IMPROVING COMPUTATIONAL PREDICTIONS OF *CIS*-REGULATORY BINDING SITES

MARK ROBINSON*, YI SUN, RENE TE BOEKHORST, PAUL KAYE, ROD ADAMS, NEIL DAVEY

Science and Technology Research Institute, University of Hertfordshire, College Lane Hatfield, Hertfordshire AL10 9AB, UK
{m.robinson, y.2.sun, R.TeBoekhorst, p.h.kaye, r.g.adams, n.davey}@herts.ac.uk

ALISTAIR G. RUST

Institute of Systems Biology, 1441 North 34th Street Seattle, WA 98103, USA
arust@systemsbiology.org

The location of *cis*-regulatory binding sites determine the connectivity of genetic regulatory networks and therefore constitute a natural focal point for research into the many biological systems controlled by such regulatory networks. Accurate computational prediction of these binding sites would facilitate research into a multitude of key areas, including embryonic development, evolution, pharmacogenemics, cancer and many other transcriptional diseases, and is likely to be an important precursor for the reverse engineering of genome wide, genetic regulatory networks. Many algorithmic strategies have been developed for the computational prediction of *cis*-regulatory binding sites but currently all approaches are prone to high rates of false positive predictions, and many are highly dependent on additional information, limiting their usefulness as research tools. In this paper we present an approach for improving the accuracy of a selection of established prediction algorithms. Firstly, it is shown that species specific optimization of algorithmic parameters can, in some cases, significantly improve the accuracy of algorithmic predictions. Secondly, it is demonstrated that the use of non-linear classification algorithms to integrate predictions from multiple sources can result in more accurate predictions. Finally, it is shown that further improvements in prediction accuracy can be gained with the use of biologically inspired post-processing of predictions.

1 Introduction

Gene regulatory networks control, to a large extent, many important biological systems, including: the accurate, and stable, expression of a subset of the proteins encoded by a genome that determine the character and properties of a cell type; the intricate program of sequential organization and subsequent cellular specialization during embryonic development; and the inherently complex dynamics of metabolic responses to pharmaceuticals. Additionally, it has become clear in recent years that much of the genetic change underlying

* Corresponding author (Mark Robinson)

morphological evolution must have occurred in gene regulatory regions [1]. To gain a functional understanding of genetic regulatory networks, along with an ability to accurately predict their topological structure and dynamics, is a research goal promising far reaching ramifications into many important biological fields.

The primary determinant of connectivity in genetic regulatory networks is the presence, or absence, of *cis*-regulatory binding sites in the regions proximal to each gene's promoter. The accurate computational prediction of the location of *cis*-regulatory binding sites is therefore a highly desirable research goal, and a key step towards the ability to reverse engineer genetic regulatory networks at a genomic scale. Such predictions could significantly streamline the, costly and time consuming, process of annotating regulatory regions by focusing attention on sequences associated with a high probability of functionality. However, prediction of *cis*-regulatory binding sites is a non-trivial problem. The rules determining which DNA sequences functionally bind transcription factors specify position dependant preferences for interactions between amino acids and nucleotide bases, rather than a simple deterministic sequence identity. In many cases, contextual information, in the form of proximally located binding sites, may play a key role in determining whether a potential binding site is in fact functional *in vivo*, further complicating computational predictions of such sites.

Many algorithms have been developed to exploit the various sources of experimental information available and the various statistical properties that appear to distinguish regulatory regions from the genome in general. These algorithms can typically be classified into four main groups based on the approach to the problem. *Scanning algorithms* attempt to generate a model, such as a position weight matrix, for each binding site from available experimental data. These models can then be used to scan potential regulatory sequences for good matches to the model. *Statistical algorithms* typically attempt to detect motifs that are considered statistically unlikely in the context of a model of the background base-pair distribution. *Co-regulatory algorithms* rely on the hypothesis that genes clustered on the basis of their expression profiles are likely to be regulated by the same transcription factors. Iterative techniques, such as Expectation Maximization, are used to generate and refine predictive models for the most over-represented motifs in the set of upstream sequences for such gene clusters. *Phylogenetic algorithms* exploit the conservation of functional DNA sequences against the background of random mutational noise during evolution. Homologous regulatory sequences from appropriately related species are compared and significant sequence alignments are predicted to act as functional *cis*-regulatory binding sites.

In spite of the wealth of research performed in the area of binding site

prediction, and the many insights gained, the current state of the art in this area is still far from perfect. In fact, results presented in this study agree strongly with other assessments of the performance of prediction algorithms [2], in showing that typically 70-80% of predictions are false positives. Interpretation of such results to guide the experimental analysis of gene regulatory regions, or the modeling of gene regulatory networks, is a difficult problem, further exasperated by technicalities of choosing an appropriate algorithm given the available data, and subsequent selection of appropriate algorithmic parameters. The utility of algorithms that scan for putative binding sites using experimentally determined weight matrices, or that require knowledge about the identity of co-regulated sets of genes, are obviously of limited use for exploring systems where that data is not available. The study of such systems is often limited to the statistical class of algorithms; statistical algorithms are, unsurprisingly, typically unable to achieve the levels of prediction accuracy observed with other classes of algorithms.

In this paper it is demonstrated, firstly, that algorithm performances can be improved by species specific optimization of algorithmic parameters. Secondly, that integration of multiple sources of algorithmic predictions, using non-linear classification techniques, can significantly improve prediction accuracy while at the same time circumventing the experimental data dependences. Finally, in order to ensure that the integration process produces biologically feasible predictions, it is necessary to perform some post-processing, and we show that this step can further improve prediction accuracy.

2 Methods

2.1 Description of the Data

Generation of appropriate data sets for use in evaluating the performance of binding site prediction algorithms is a challenging problem with no clear solution [2]. The use of promoter sequences that have been experimentally annotated is commonly used, although with no assurance of the completeness of sequence annotations, penalization of some correctly predicting algorithms is inevitable. An alternative strategy is for the stochastic generation of random sequences embedded with examples of binding sites, but, our current lack of knowledge of the stochastic processes underlying the sequences found in nature renders this strategy open to unknown biases.

For the purposes of this study we chose the annotated sequence strategy, attempting to minimize the error by using promoter sequences from one of the

most well studied model organisms, the *S.cerevisiae* promoter database[†]. 120 annotated promoter sequences were selected for training and testing the algorithms, a total of 68910 bp of sequence data. In addition, homologous promoter sequences for 59 of the sequences were collected from *S.paradoxus*, *S.mikatae* and *S.bayanus*, and 69 of the sequences were determined, by the use of micro-array studies, to be likely candidates for co-regulation.

For integration of multiple algorithmic predictions a matrix was generated, consisting of 68910 12-ary real valued vectors, each associated with a binary label indicating the presence or absence of a binding site annotation at this sequence position. Each 12-ary vector represents the predictions, at this position in the sequence dataset, for each of the twelve algorithms. All predictions are normalized as real values in the range [-1, 1] with 0 allocated to sequence positions where algorithm predictions were not possible.

In this work we divided our dataset into a training set and a test set: the first 2/3 for training and the final 1/3 for testing. Additionally, we contextualize the training and test datasets to ensure that the classification algorithms have data on contiguous binding site predictions. This is achieved by windowing the vectors. We use a window size of 7, providing contextual information for 3 bps either side of the position of interest. This procedure carries the considerable benefit of eliminating a large number of repeated or inconsistent vectors which are found to be present in the data and would otherwise pose a significant obstacle to the training of the classifiers.

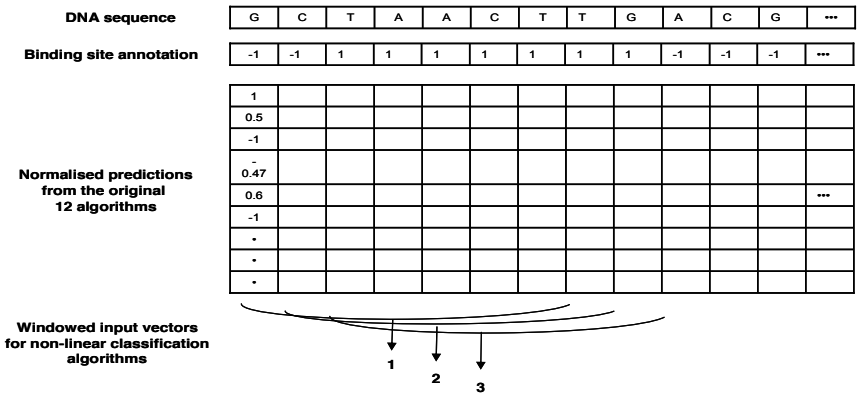

Figure 1: Organization and structure of the dataset used to train classification algorithms. Windowing of the input data is show, for example, the first input vector consists of the first seven columns of predictions concatenated together. The target output is the binding site annotation corresponding to the middle column of the window (i.e. the fourth column for input vector 1)

[†] SCPD: http://cgsigma.cshl.org/jian/

2.2 Performance Metrics

Approximately 7% of our dataset, consisting of 68910 data points, are labelled as annotated binding sites, making this an *imbalanced* dataset [3]. Supervised classification algorithms would be expected to over predict the majority class in an imbalanced dataset, i.e. in this instance a success rate of 93% could be achieved by only predicting the majority class, namely the non-binding site class. In this work we deal with this issue in two ways: firstly, with the use of appropriate metrics for evaluation of algorithmic performance and secondly, with the use of data-based methods during classifier training (see 2.5).

Several common performance metrics such as *Recall*, *Precision* and *F-score* [4] are defined to allow evaluation of performance on the minority class. The *Correlation Coefficient* [2] is also defined, providing a measure of the correlation of predicted binding sites to the annotated data. Each of these metrics is defined using a confusion matrix (see Table 1):

Table 1: A confusion matrix – TN is the true negative count, FP is the false positive count, FN is the false negative count and TP is the true positive count.

TN	FP
FN	TP

$$\text{Recall} = \frac{TP}{(TP + FN)}, \quad \text{Precision} = \frac{TP}{(TP + FP)}$$

$$\text{F-Score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}, \quad \text{FP-rate} = \frac{FP}{(FP + TN)}$$

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

2.3 Description of Prediction Algorithms Used

The binding site prediction tools evaluated in this study were selected, either from the research literature or as tools developed in-house (PARS & DREAM) or by collaborators (Sampler), to include representatives for all the major prediction strategies, see Table 2. The aim in selecting these disparate algorithms was to maximize the relevant information with the full set of binding site predictions. Where possible, algorithmic parameters were set to those reported in the literature; for the remainder default parameter settings were used.

Table 2: Categorization of algorithms used in study

Strategy	Algorithm
Scanning	Fuzznuc[‡] Motif Scanner [5] Ahab [6]
Statistical	PARS[§] Dream (over and under represented motifs) Verbumculus [7]
Co-Regulatory	MEME [8] AlignACE [9] Sampler[**] (Institute for Systems Biology)
Evolutionary	SeqComp [10] Footprinter [11]

2.4 Species Specific Optimization of Prediction Algorithm Parameters

The many algorithms available for *cis*-regulatory binding site prediction have typically been developed, and suitable operating parameters selected, for a specific model organism. It is an open question as to whether such operational parameter settings would be expected to be optimal across a wide range of organisms, although in practice this is often the assumption. It was decided to search the parameter space of each algorithm to find optimal settings for binding site detection in the yeast dataset.

The parameter space consisted of an assemblage of various data types: Boolean, discrete and real valued types of varying ranges. An implementation of an efficient simulated annealing schedule [12] was used to search the parameter space. All optimization runs were performed with a single algorithm and were initialized with the default parameters. Evolution of novel solutions in the parameter space was achieved by the random selection of one parameter per iteration with the subsequent selection of a new random point with the range of the selected parameter. The user-specified variable, λ [12], that determines the rate at which stocasticity deceases over time, was set to a value of 0.001. The training set was divided into two equal parts for training and validation of performance during the optimization process. The fitness function was implemented using the F-Score performance metric.

[‡] http://www.hgmp.mrc.ac.uk/Software/EMBOSS/
[§] http://sourceforge.net/projects/pars/
[**] http://sourceforge.net/projects/netmotsa

2.5 Sampling Techniques for Learning Imbalanced Datasets

To ensure efficient training of classifiers on this imbalanced dataset, data based sampling techniques [13, 14] were employed, namely under-sampling of the majority class (negative examples) and over-sampling of the minority class (positive examples). For under-sampling, we randomly selected a subset of data points from the majority class. The more complex issues that arise with over-sampling [3] are addressed by the use of *synthetic minority over-sampling* as proposed in [13]. In the absence of these sampling techniques, the supervised classifiers achieved negligible rates of true positive predictions. The number of items in the minority class is doubled and degree of under-sampling is chosen so as to ensure the final ratio of minority and majority members is one half. Preliminary cross-validation experiments were used to set these parameters, this parameter space will be explored more thoroughly in future work.

2.6 Supervised Classifiers

A variety of supervised classification algorithms were used to explore their relative merits for improvement of prediction accuracy by the integration of predictions from multiple algorithmic sources [15]. A single layer neural network (SLN) was used in this study to provide a standard for baseline performance. A Support Vector Machine (SVM), an effective, contemporary kernel based classification algorithm was utilized. The final algorithm used was the Adaboost algorithm [16], a powerful, recently proposed method for producing a strong classifier from a sequence of weak classifiers. The algorithm begins by training a weak classifier, here an SLN, on the original dataset. A new dataset is then produced by increasing the frequency of data points poorly classified. This process then iterates until a strong classifier has been produced.

2.7 Biologically Constrained Post-Processing

Observation of the predictions from the supervised classifiers used here, suggest that many of their false positive predictions could be ruled out based on known, or suspected, biological constraints of functional binding sites. One possible constraint is that a binding motif must be of sufficient length to make randomly occurring copies unlikely. Predictions that fall below some threshold length are therefore prime candidates for post-processing, either to filter them out, or to extend their size. This is a particularly pertinent step as the meta-predictions generated from the original, noisy, algorithmic predictions can produce fragmented predictions as an artefact of the integration process. In this study a post-processing step is incorporated filtering out predictions that do not reach a

minimum threshold for contiguous length. Classification performance is evaluated for threshold values of 5 bp and 6 bp, as shown in Section 3.3.

3 Results

3.1 Comparison of Performance Using Default and Species Specific Optimized Parameters

Each of algorithms used in this study were initially evaluated on the annotated *S.cerevisiae* sequence test set of 22967 bp, producing a set of scores using their respective default parameters. These scores were used as a baseline for the evaluation of performance using parameter sets identified as conferring a performance improvement during training on the *S.cerevisiae* training set of 42919 bp. It is important to note that performance was evaluated over the entire test dataset; an algorithm is effectively penalized when it is unable to make predictions for specific sequences due to lack of supplementary data. When evaluated on the subset of sequences where predictions were made, MEME, for example was able to achieve an optimized F-Score of over 45%, although, when evaluated on the entire dataset its performance dropped to an F-Score value of 18.21%, as shown in Table 3. However, as we are interested in evaluating the functional usefulness of the algorithms, with an aim to overcome these limitations by integrating multiple sources of information, the full test dataset is most appropriate.

Table 3: A comparison of algorithmic performance using default vs. optimized parameters. Dashes indicate that no improved parameters settings could be found or that optimization was not possible.

Algorithm	**Default**			**Optimized**		
	F-Score	**CC**	**FP-rate**	**F-Score**	**CC**	**FP-rate**
Fuzznuc	24.59	19.02	10.61	-	-	-
PARS	10.41	2.42	12.45	-	-	-
Verbumculus	19.24	12.79	12.23	-	-	-
Ahab	14.31	7.45	48.86	25.60	22.59	3.36
Dream (over)	9.21	-1.02	24.13	-	-	-
Dream (under)	8.25	-2.58	25.19	13.40	5.54	15.53
Motif Scanner	16.19	9.41	9.95	-	-	-
Sampler	4.72	1.19	2.56	6.86	11.72	3.14
MEME	18.21	15.61	2.43	18.83	45.23	0.94
AlignACE	13.09	11.08	2.03	-	-	-
SeqComp	9.56	2.11	9.81	-	-	-
Footprinter	10.39	3.19	9.20	-	-	-

F-Score performance, with default parameters, was typically below 20% with even lower scores for the correlation coefficient. The simple scanning algorithm, Fuzznuc, easily outperforms the others with an F-score of nearly 25%. The last 5 algorithms in Table 2, the co-regulatory and phylogenetic algorithms, produced notably low FP-rates. This is likely due to their predictions, where possible, being of high acuity but of a conservative nature.

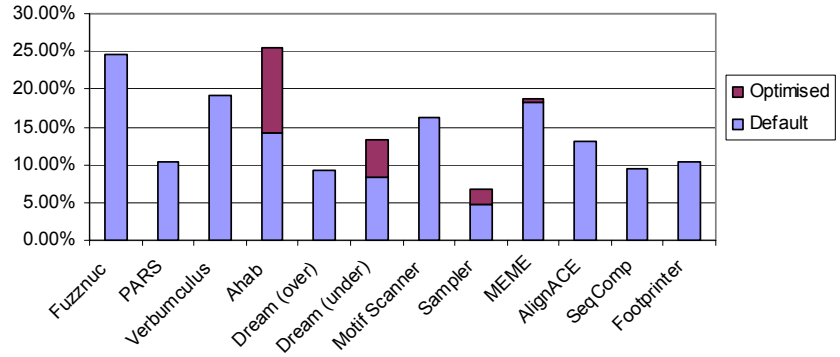

Figure 2: Comparison of F-score performance of algorithms using default and optimized parameters

Optimized parameters were found for Ahab, Dream (over), Sampler and MEME that improve performance on the test set, as can be seen in Figure 2. The performance improvement seen with Ahab was particularly impressive, with an 80% increase in F-score while the false positive rate was reduced by 93%. It is intriguing to note that both Ahab, and Dream, were developed using *D.melanogaster* as a model. Conversely, for Verbumculus and AlignACE, both known to have been developed with *S.cerevisiae* as a model, no parameter improvement could be found. The possibility is certainly raised that species specific parameter optimization may be necessary for optimal algorithmic predictions; it remains to be seen whether this situation will in fact prove to be the case for other organisms and if so what the underlying causes in terms of *cis*-regulatory organization and structure might be.

3.2 Integration of Multiple Algorithmic Predictions Using Supervised Classifiers

Another important question is whether performance can be further refined by the integration of multiple algorithm predictions. To this end, three non-linear supervised classifiers were trained using the predictions of the original algorithms on the training set. Cross-validation was used to select appropriate parameter settings for the classifiers, with each parameter setting being trained on 4/5 of training set and validated on the final 1/5. The algorithm, Fuzznuc,

was chosen to provide a baseline performance based on its high performance across the entire test dataset. The optimized version of Ahab, which achieved even higher levels of performance, was not available in time to be included in this study.

Table 4: Comparison of Fuzznuc performance vs. integration strategies performance

Algorithm	F-Score	CC	FP-rate
Fuzznuc	24.59	19.0	10.1
SLN	25.0	19.5	7.3
SVM	27.2	21.8	8.1
Adaboost	27.0	21.7	6.3

The results in Table 4 show a clear and consistent picture. Integration of multiple algorithm predictions consistently results in more accurate predictions, as measured by the F-Score, correlation coefficient and FP-rate. The SVM out performs all other algorithms, improving on the F-Score performance of Fuzznuc by 10%, the correlation coefficient by 14% while reducing the FP-rate by 38%.

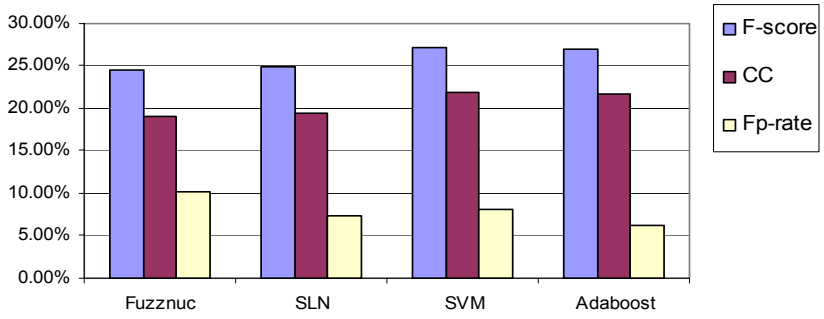

Figure 3: Performance statistics comparing the accuracy of different classification strategies

3.3 Refinement of Results Using Biologically Constrained Post-Processing

The final step in our refinement of binding site predictions is the conceptually simple one of ensuring that all predictions are biologically viable. Table 5 details the results of an experiment designed to explore whether small, fragmented predictions were artefacts of the meta-analysis shown in Section 3.2.

Table 5: Performance improvements with a range of minimum word size filter thresholds

Algorithm	No Filtering			Filter < 5			Filter < 6		
	F-Score	CC	FP-rate	F-Score	CC	FP-rate	F-Score	CC	FP-rate
SLN	25.0	19.5	7.3	25.6	20.3	5.9	26.0	20.8	5.5
SVM	27.2	21.8	8.1	28.2	23.0	6.7	28.4	23.2	6.2
Adaboost	27.0	21.7	6.3	28.0	23.2	4.7	27.3	22.8	4.3

It can be seen that in all cases filtering out predictions less than 5 bp in extent, improves prediction accuracy, by all performance measures, for all classifiers. Filtering predictions that are less than 6 bp, further improves performance for both the SLN and SVM but causes a considerable drop in accuracy for the Adaboost algorithm. In the best case, the combination of integration using the SVM combined with the post-processing filtering with threshold < 6 improves the F-Score performance relative to Fuzznuc by 15%.

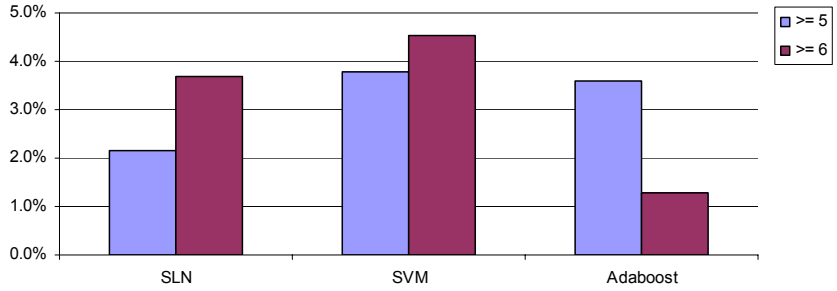

Figure 4: Percentage change in F-Score, relative to unfiltered results, after post-processing of fragmented results using thresholds 5 and 6 respectively

4 Conclusions

The important, and significant, result presented here is that an incremental approach to algorithmic refinement can produce considerable improvement in prediction accuracy. In the best case, the combination of integrating predictions using the SVM followed by post-processing filtering using a threshold < 6, improves the F-Score to 28.4%, an improvement of 15% relative to the performance achieved by the best algorithm, Fuzznuc.

The performance improvements achieved by parameter optimization, most notably those of Ahab, are highly suggestive; optimal computation prediction of cis-regulatory binding sites may require species specific optimization of parameter sets.

The use of supervised classification techniques for integrating predictions from multiple sources is shown to be a particularly promising approach. The success of integrating these multiple prediction sources indicates that there is additional information to be exploited, collectively, in these prediction sets.

Initial attempts at post-processing meta-predictions were worthwhile and present many opportunities for future work in this area. Other important biological constraints that might be explored in future work include, clustering of predicted sites, and bias in the base pair distributions within predicted sites.

References

1.Davidson, E.H., *Genomic Regulatory Systems: Development and Evolution*. 2001, San Diego: Academic Press.
2.Tompa, M., et al., *Assessing computational tools for the discovery of transcription factor binding sites*. Nat Biotechnol, 2005. **23**(1): p. 137-44.
3.Japkowicz, N. *Class imbalances: Are we focusing on the right isse?* in *Workshop on learning from imbalanced datasets, II, ICML*. 2003. Washington DC.
4.Bussemaker, H.J., H. Li, and E.D. Siggia, *Regulatory element detection using correlation with expression*. Nat Genet, 2001. **27**(2): p. 167-71.
5.Thijs, G., et al., *A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling*. Bioinformatics, 2001. **17**(12): p. 1113-22.
6.Rajewsky, N., et al., *Computational detection of genomic cis-regulatory modules applied to body patterning in the early Drosophila embryo*. BMC Bioinformatics, 2002. **3**(1): p. 30.
7.Apostolico, A., et al., *Efficient detection of unusual words*. J Comput Biol, 2000. **7**(1-2): p. 71-94.
8.Bailey, T.L. and C. Elkan, *Fitting a mixture model by expectation maximization to discover motifs in biopolymers*. Proc Int Conf Intell Syst Mol Biol, 1994. **2**: p. 28-36.
9.Hughes, J.D., et al., *Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae*. J Mol Biol, 2000. **296**(5): p. 1205-14.
10.Brown, C.T., et al., *New computational approaches for analysis of cis-regulatory networks*. Dev Biol, 2002. **246**(1): p. 86-102.
11.Blanchette, M. and M. Tompa, *FootPrinter: A program designed for phylogenetic footprinting*. Nucleic Acids Res, 2003. **31**(13): p. 3840-2.
12.Lam, J. and J. Delosme, *Performance of a New Annealing Schedule*. Proceedings 25th ACM/IEEE Design Automation Conference, 1988: p. 306-311.
13.Chawla, N.V., et al., *SMOTE: Synthetic minority over-sampling Technique*. Journal of Artificial Intelligence Research, 2002. **16**: p. 321-357.
14.Radivojac, P., et al., *Classification and knowledge discovery in protein databases*. J Biomed Inform, 2004. **37**(4): p. 224-39.
15.Sun, Y., et al. *Integrating binding site predictions using non-linear classification methods*. in *Machine Learning Workshop*. 2005. Sheffield: LNAI.
16.Freund, Y. and R.E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of Computer and Systems Sciences, 1997. **55**(1): p. 119-139.