Promoter Prediction Using Physico-chemical Properties of DNA

Philip Uren, R. Michael Cameron-Jones, Arthur Sale

School of Computing, Faculty of Science, Engineering and Technology,
University of Tasmania, Hobart and Launceston, Tasmania, Australia.
[Philip.Uren, Michael.CameronJones, Arthur.Sale] @utas.edu.au

Abstract. The ability to locate promoters within a section of DNA is known to be a very difficult and very important task in DNA analysis. We document an approach that incorporates the concept of DNA as a complex molecule using several models of its physico-chemical properties. A support vector machine is trained to recognise promoters by their distinctive physical and chemical properties. We demonstrate that by combining models, we can improve upon the classification accuracy obtained with a single model. We also show that by examining how the predictive accuracy of these properties varies over the promoter, we can reduce the number of attributes needed. Finally, we apply this method to a real-world problem. The results demonstrate that such an approach has significant merit in its own right. Furthermore, they suggest better results from a planned combined approach to promoter prediction using both physico-chemical and sequence based techniques.

Key words: promoter prediction, support vector machine, SVM, physico-chemical, classifier, DNA, transcription.

1 Introduction

In-silico eukaryotic promoter recognition is known to be a difficult problem [1]. Objective statements about the current state of the art in terms of promoter recognition are complicated by the wide selection of metrics used for assessing performance. Moreover, an optimal trade off between sensitivity and specificity is not immediately apparent. In some applications, a high sensitivity is valued – i.e. it is best to find as many actual promoters as possible and a relatively high false positive rate is tolerable. In contrast, in other situations the specificity may be more important, particularly where it is expensive to validate predictions. As an indication of the progress to date in the field, Bajic, Tan et al. [2] report that none of the programs they tested achieved a combined sensitivity and specificity greater than 65%.

The purpose of this paper is to examine how effectively physico-chemical properties of DNA can be used to predict the location of promoters within the human genome. Previous studies have demonstrated that promoters exhibit distinct patterns in terms of these properties [3-7]. Physical properties of DNA have also been shown to be important in terms of a biological understanding of the mechanisms of transcription – for example [8-10]. Approaches to separating promoters from non-promoters in E-coli using physico-chemical properties have met with success [11, 12].

Ohler, Niemann et al. proposed an approach for incorporating them into their promoter recognition program, McPromoter [13]. They demonstrated that this reduced the false positive rate on a given test set by about 30%. In contrast to the approach we employ within this work, they computed the mean value for a given model within a segment of the instance. The segmentation was based upon the sequence alone.

It has also been shown in previous work that an encoding of sequence data using structural models can be more efficient than a sequence based encoding. Using a single model Baldi, Chauvin et al [14] demonstrated that similar accuracy to a sequence based approach was possible but with only about a fourth of the attributes required. We explore a different approach to reducing the size of the representation.

In their important examination of the application of physico-chemical properties to the clustering and classification of promoters, Florquin, Saeys et al. [4] examined how effectively certain properties could be used to discriminate between promoters and non-promoters. We aim to extend their exploration in several significant directions. Firstly, they examined the application of a single model at a time. We explore the application of multiple models simultaneously and assess their classification performance, demonstrating an improved accuracy over any single model. Secondly, we examine which models are of importance within which segments of the promoter. We use this information to demonstrate how comparable accuracies can be achieved with fewer attributes, reducing computational time. Finally, we explore the application of this approach within a more realistic scenario – the classification of a contiguous segment of human DNA from chromosome 21, of length approximately 10Mbps. We measure the results by means of the approaches used by Bajic, Tan et al. [2] and show that this technique has merit in its own right for promoter prediction

To the authors' knowledge, this is the first large-scale application of a promoter prediction method that uses *only* physico-chemical properties. We demonstrate that they are actually quite effective at picking up promoters on their own. Although we are not advocating an abandonment of sequence related techniques, we present these results as further impetus to re-examine the abstraction of DNA (and biological information in general). That is, DNA is often represented in computational areas as simply a string of characters. Results such as those presented within this paper suggest that its properties as a complex molecule are not only useful, but essential for various forms of computational modelling and biological understanding.

2 Materials and Methods

2.1 Datasets and Physico-chemical Properties

We make use of the publicly available DBTSS (which can be accessed at: http://dbtss.hgc.jp/) for the location and sequence data of human promoters [15-17]. For training the classifier, we used this dataset as the positive instances and randomly selected an equal number of negative instances from the human genome. When testing on chromosome 21, we used a modified version of this dataset for training, which excluded all data from chromosome 21.

Based upon the recommendations of Florquin, Saeys et al. [4], we selected six models for describing the physico-chemical properties of the DNA sequences. These were: A-philicity [18], DNA bending stiffness [19], DNA denaturation [20], duplex disrupt energy [21], nucleosome position preference [22] and propeller twist [23].

When training the classifier, all sequence data is already in uniformly sized instances (we use an instance size of 150bp with 100bps downstream and 50bps upstream of the TSS). From the raw sequence information, we evaluate each of the models listed above. This produces 6 sequences, of size 149 or 148 for a di- or tri-nucleotide model respectively, which we then smooth with a window of size 10 and step of 1. After smoothing, the sequences are of length 139 or 140. These 6 sequences are then concatenated and represent a single instance as presented to the classifier.

When scanning a contiguous segment of DNA for promoters, we first split the sequence into segments of 150bps in size, using a step of 1. That is, if the sequence is *n* bps in size, we convert this to *n-150* segments of 150bps in size. Put another way, each segment differs from its nearest neighbour by only 2 bps (one at each end). The processing of these segments then proceeds as described above.

When applying attribute pruning, we train the classifier on the dataset as described above and examine the weights associated with each of the attributes. Attributes with low weights are then removed from the dataset. In some cases we prune all attributes with a weight below a certain threshold. In other cases the top *n* weighted attributes are retained and the remainder are pruned. We specify which of these two approaches are being employed when we present the results.

2.2 Classifier and Post-processing

We employ a support vector machine using a linear kernel, specifically the implementation present in WEKA [24-26] using the default settings, to classify instances as either promoter or non-promoter. We also post process the output of the support vector machine. The motivation for this is explained in more detail within the results section. The approach taken here is to locate runs of positive predictions within a window of a certain size and exceeding a threshold of a given percent of positives. For all the results presented within this paper, we set this to be a window of size 400bps containing 85% positives or more. Different values for these parameters result in different balances of sensitivity and specificity. This produces a window of varying length describing the promoter region. For the purposes of comparing TSS prediction position, we examined taking the start, middle and end of this window. Given the metrics we are using, we determined that taking either the middle or end produced the same results, but taking the start was markedly worse.

Due to the fact that training times are quite long for some of the approaches we present herein (most notably the combined model with 839 attributes), we do not perform a ten-fold cross validation as is often done. Instead, we present the result of training the model on increasing segments of the data (from 1% up to 50%) and show the result of testing on the left over data (from 99% down to 50%). In each case the sets are cumulative. That is, the 5% dataset contains all of the 1% dataset; the 10% dataset contains all of the 5% and the 1%, etc.

Using the same approach as Bajic, Tan et al. [2], we count a true positive (TP) when a prediction falls within 2000bps of a TSS. We also aggregate predictions that are within 1000bp of each other. In evaluating the performance of the classifier we make use of five metrics throughout the paper. These are *Sensitivity* – abbreviated to Se (the percentage of positives that are correctly identified), *Positive Predictive Value* – abbreviated to *PPV* (the percentage of positive predictions that are correct), *Accuracy* (the percentage of predictions that are correct, be they negative or positive), *True Positive cost* (the number of FPs required to achieve a TP) and finally *specificity* – abbreviated to Sp (the percentage of negatives correctly identified). Expressed more succinctly, Se = TP/(TP+FN), PPV = TP/(TP+FP), Accuracy = (TP+TN)/(TP + FP + FN + TN), TP cost = FP / TP, and Sp. = TN/(TN+FP).

3 Results and Discussion

3.1 Single vs. Combined Model

Within this section, we examine the results of comparing the classification performance of a single model with that of a combined model, using the six models suggested as highly discriminative by Florquin, Saeys et al. [4]. We use the full promoter dataset, as described in the section materials and methods. The results are summarised in Table 1, using a simple accuracy (i.e. percentage of correct predictions) metric.

As can be seen, there is an approximately 2% increase in accuracy achieved by the combined model over the best of the single model accuracies. Although the difference is not great, the 2% improvement would account for hundreds of instances when considered from the perspective of a genome wide scan, or similar magnitude. This is clearly a significant improvement, although it comes at the cost of computational time. Using half the dataset for training (approx 8,000 instances), a single model SVM takes only approximately ten minutes to train. In contrast, the combined model,

Table 1. The accuracy of the SVM in separating promoter sequences from non-promoter sequences in a 50% positive/negative dataset. Promoter sequences are taken from DBTSS and non-promoter sequences randomly selected from the human genome

Training set proportion (%)	1	5	10	15	20	30	40	50
A-philicity	81.4	84.8	84.7	85.0	85.4	86.1	86.5	86.6
DNA Bending Stiffness	82.9	84.8	84.8	84.8	85.1	85.6	85.9	86.2
DNA Denaturation	80.4	84.2	84.5	85.0	85.1	85.5	86.0	86.2
Duplex Disrupt Energy	85.1	86.9	87.5	87.7	87.9	88.4	88.7	88.7
Nucleosome Positional Pref.	83.2	86.5	86.5	86.7	87.0	87.2	87.3	87.7
Propeller Twist	85.1	87.0	87.3	87.5	87.6	88.0	88.3	88.4
COMBINED MODELS	81.5	87.4	88.3	88.9	89.3	90.1	90.4	90.7

using the same size training set takes approximately two hours. Putting this in perspective, however, the SVM need only be trained once and the difference in classification time between the two approaches is less of an issue due to the relatively small figures (approx. 8.15s and 0.35s respectively for slightly more than 8,000 instances).

These two approaches allow a trade off between speed and accuracy. Within the next section, we will examine a third technique that can equal the accuracies of the single model approach but uses less than half the number of attributes, resulting in a further speed improvement.

3.2 Attribute Pruning

Due to the characteristics of the learning method we are using (i.e. an SVM), we can examine the weights associated with the attributes used. As described in the section *materials and methods*, input is provided to the SVM as a series of instances. Each instance describes the result of applying the given models to the sequence data returned by a sliding window of 150bps. We will now examine these weights to determine which parts of the promoter are discriminative in terms of each model.

We graph these attribute weights in Figure 1. As can be seen, the area around the TSS is of most discriminatory power, with many models displaying clear spikes in this region (150bps are shown for each model; the TSS is at the 100th bp).

Curiously, A-philicity has the highest weighted attribute spike but when used on its own (as discussed in the previous section and shown in Table 1) it does not produce the best classification performance. This seems to indicate that A-philicity is predictive of promoter activity when combined with other models, but less so when considered independently. In contrast, Duplex Disrupt Energy clearly performs well on its own and is highly influential in a combined approach.

We applied the attribute pruning describe in the section *materials and methods* to both single models and the combined model presented in the previous section. The results are shown in Table 2. These datasets are described by only 61 attributes, as

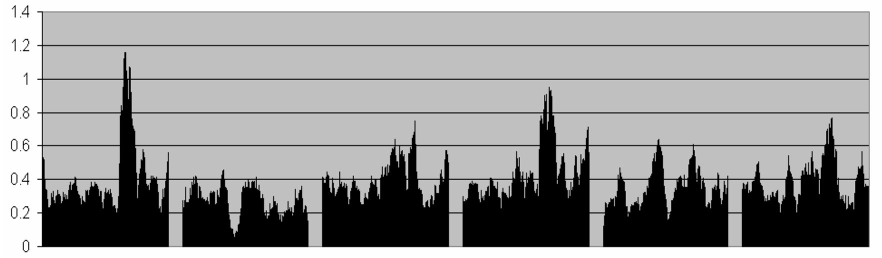

Fig. 1. The attribute weights associated with each of the 839 attributes used in the combined model approach. The weights are segmented according to model in the following order: A-philicity, DNA bending stiffness, DNA denaturation, duplex disrupt energy, nucleosome positional preference and propeller twist.

Table 2. The accuracy of the models after attribute pruning. The combined model was pruned to include only those attributes weighted higher than 0.6 – 61 attributes. The single models were pruned to include the 61 attributes with the highest weights

Training set proportion (%)	1	5	10	15	20	30	40	50
A-philicity	82.6	84.9	85.2	86.0	85.7	86.3	86.5	86.6
DNA Bending Stiffness	83.5	84.9	84.8	84.9	85.3	85.6	85.9	86.1
DNA Denaturation	82.2	84.3	85.1	85.1	85.6	85.7	85.9	86.0
Duplex Disrupt Energy	85.8	87.1	87.7	87.9	88.1	88.5	88.7	88.8
Nucleosome Positional Pref.	84.5	86.7	86.8	86.8	87.1	87.3	87.5	87.7
Propeller Twist	85.8	87.5	87.5	87.7	87.8	88.1	88.2	88.5
COMBINED MODELS	83.0	86.0	86.3	86.6	86.8	87.1	87.3	87.5
DDE + PROP. TWIST	86.7	88.1	88.3	88.4	88.6	89.0	89.2	89.3

compared to the 139/140 (for a di- or tri-nucleotide model) attributes required for a single model and the 839 attributes required for the combined model. Despite the significantly smaller size, one can see that performance is comparable to evaluating each model over the entirety of the instance. The combined model loses enough accuracy from the pruning to no longer be the best choice under the simple accuracy metric used here. We also present an alternate approach to pruning, listed as DDE + prop. twist. This method first prunes the duplex disrupt energy and propeller twist models to 30 attributes and then combines them to make a model of 60 attributes. As can be seen, this approach produces the highest accuracy. The training time required for these pruned models is roughly half that of the full models, at approximately 5 minutes. Most importantly though, the classification time on approximately 8000 instances is only 0.17s. With such a small number, classification time is now insignificant in comparison to the time required for operations such as file loading. On large datasets, this fast classification time is invaluable.

To briefly summarise, we have demonstrated that a combined approach using multiple physico-chemical properties improves accuracy over any single model considered on its own. We have also shown that if speed is more important than accuracy, these models can be evaluated over a smaller portion of the input data for an almost negligible lose of accuracy but a significant improvement in execution time.

3.3 Real World Performance

As mentioned in the previous section, the results obtained from applying these methods to discriminating between isolated promoter and non-promoter instances are encouraging. However, it is not yet known whether they can be extrapolated to more realistic uses. We now apply the two combined model SVMs produced in the above section to contiguous sections of sequence data, taken from human chromosome 21. Note that when training the SVMs we used no data from chromosome 21.

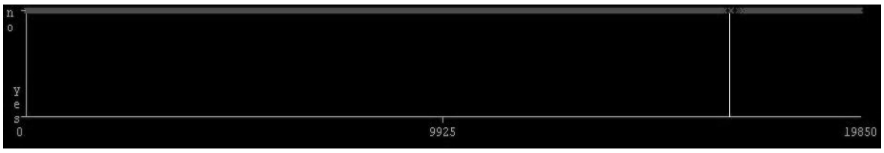

Fig. 2. The classification diagram produced by the SVM. The X-axis is the position within the input sequence data. The top line represents the locations of negative predictions. Notice the clear gap in negative predictions centred on the TSS position (indicated by the white line)

We took 25 promoters from the first half of chromosome 21 and extracted 20,000 bps around each one producing 25 testing sets. We found that both SVMs on average labelled 20% of the dataset (or approx 4000 instances) as being promoters.

Although this is a poor result, we observed that the distribution of predictions was correlated with the location of actual positives. This is best illustrated with a diagram and an example is presented in Figure 2. As can be seen, the actual location of the TSS coincides with a gap in the prediction of negatives by the classifier. We used this observation as the basis for a second level to the classifier, which searches for these gaps in the negative output. The results obtained using the 25 segments from the first-half of chromosome 21 were used to select a size for the gap and the percentage of instances within the gap that must be positive. We also determined that taking the middle or the end of a gap as the predicted TSS was equally accurate, but taking the beginning produced significantly worse results.

3.4 Combined Classifier Performance

We now examine the performance of the combined classifiers, using a section of DNA from the second half of chromosome 21 of about 10Mbps. Interestingly, we found that the SVM produced from taking a 0.6 cut-off to the full combined model produced fewer false positives than the most highly weighted attributes from the duplex disrupt energy and propeller twist models. This in turn leads to a worse PPV for the DDE + Prop Twist model. It does however have one advantage: producing the highest sensitivity (at 66%) of the two models examined, although this comes at the cost of a very low PPV (approx. 8%). It is not clear whether the other combined model might also produce sensitivity within this range if more relaxed parameters allowed a lower PPV. Because of space restrictions, we choose to omit a detailed comparison and focus our attention on the model labeled “combined model” above. We feel that both the higher PPV values and the ability to provide a better balance between the PPV and Se. make this model more practical in a realistic environment.

Within the region we have chosen for testing there are 65 promoters and the classifier correctly identifies 36 of them (a sensitivity of 55.4%). In total, 224 positive predictions are made, equating to a positive predictive value (PPV) of 16.1%. The sensitivity is comparable to many other current approaches for promoter prediction – of the 14 programs examined by Bajic, Tan et al [2] 9 of them score sensitivity of approximately 55% or less. However, we are aware that the low PPV value is a drawback to the proposed method and now suggest an approach for improving it.

As we have mentioned previously, the output from the upper level of the classifier is a window of predicted promoter activity. We now consider the size of the window as an indication of confidence in respect to the prediction. This leads to the ability to discard low confidence predictions. Because we are aggregating windows that are within 1000bp of each other, this allows for two possible approaches in respect to applying a window-size threshold; before aggregation and after. We present the results obtained from both approaches graphically in Figure 3.

The most striking feature is that by applying a threshold of 1200, 1300 or 1400 bp to the window size before aggregation, we can achieve a PPV of greater than or equal to 70%. However, this comes at the cost of sensitivity, which falls to less than 15%. Whether this trade-off is worth it depends entirely on perspective: if the cost of false positives is paramount, then the answer may well be yes. It is also interesting to note that by applying a threshold of about 300bp, we can achieve an improvement in PPV for an almost negligible loss of sensitivity. A reasonable trade off is also apparent with a threshold of about 500bp, where PPV and Se of approximately 40% is achieved. It is also apparent that thresholding before aggregation favours PPV whereas after aggregation favours sensitivity. Intuitively, this is as one would expect. Because of the nature of the aggregation, it is possible that, for example, two small windows which are 900bps from each other may be aggregated into one large window of more than 1000bps. Hence, thresholding before aggregation favours large uninterrupted windows, which are more likely to represent actual promoters, while discarding the interrupted windows which are less likely.

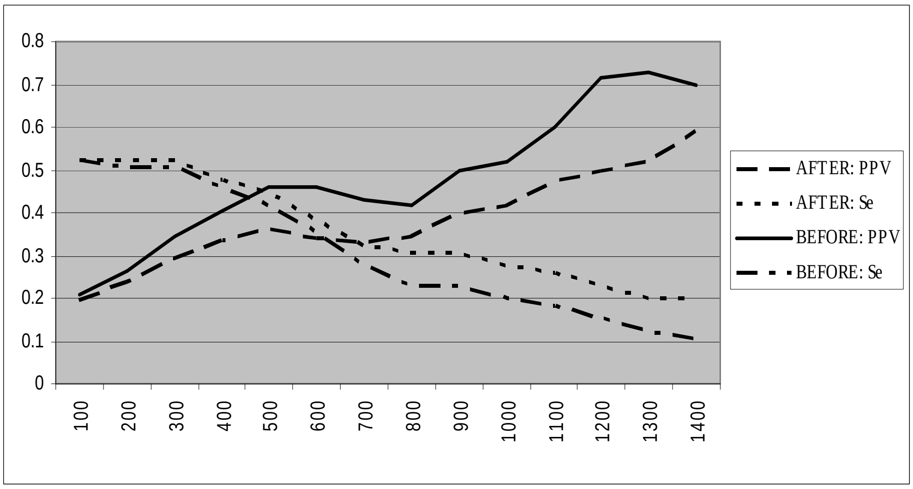

Fig. 3. The performance of the classifier when applying a post-processing step of discarding small windows. The window threshold is shown on the x-axis. We graph both sensitivity and positive predictive value for threshold application both before and after aggregation.

4 Conclusions

We have set out within this paper to address two distinct points. Firstly, we intended to demonstrate that a combined model of physico-chemical properties could be used for promoter prediction, improving the accuracy of using a single model. We have demonstrated this fact, showing an increase of approximately 2% over the best of the single models. We stress once again the importance of such a result in considering large datasets, or within applications where sensitivity is of importance.

We have also explored the relative importance of the different properties in different segments of the promoter in terms of predictive power by examining the attribute weighting of the support vector machine produced. Using these results, we produced a reduced model, which is faster but loses little in terms of accuracy. These two results combined demonstrate that this approach can be tailored for sensitivity by using the full combined model or for speed by using attribute pruned models.

The second component of this paper has looked at how these approaches perform when confronted with large, real world problems. We showed that although the raw results of the SVM do not immediately appear encouraging, by post-processing the output, it is possible to predict promoter locations with a sensitivity of 55.4% and a PPV of 16.1%. By using a threshold approach to the window size, we also demonstrated that a balanced 40% PPV and Se. was possible. Further to this, by modifying the threshold size, the approach can be tailored for either sensitivity or specificity. These results demonstrate that a promoter prediction technique based only on physico-chemical properties is possible and capable of performance that is competitive with established approaches. The use of the physico-chemical properties of DNA for promoter prediction is a promising direction for further research, both on its own and in conjunction with sequence based methods.

5 Further Work

There are several distinct directions to pursue as a result of this work. Firstly, Florquin, Saeys et al. [4] suggest that promoters be treated as distinctly separate groups. Within this work, we have not implemented this idea, treating all promoters as being the same. By learning separate classifiers for distinct types of promoters, improvements in sensitivity and PPV may be possible. Alternately, a simpler approach of retraining the classifier using the instances that were not correctly classified on the first pass may produce equivalent performance. There is also the question of how to combine these multiple classifiers.

Bajic, Tan et al. [2] also suggest that the masking of repeats may improve the performance of promoter prediction programs. They also present evidence to suggest that experiments run on single chromosomes may not be representative of results produced on the whole genome. Future work on this approach would require the exploration of both these ideas: the application of repeat masker and larger-scale tests.

Furthermore, we have presented proof that combined models can outperform single models, but have only examined a few possible combinations. Our results indicate that the combination of models may not be as simple as originally thought and a more thorough exploration is called for.

Finally, we have shown that the application of physico-chemical properties to the problem of promoter prediction can produce competitive results on their own, but we are not advocating the abandonment of other approaches. A combined approach, employing these techniques and other complementary (possibly sequence related) approaches may produce better results. In particular, this approach is currently not strand specific. That is, predictions are equally likely to be on either strand. By employing sequence based techniques, this could be overcome, allowing a strand specific prediction to be made.

Acknowledgements

This work was supported by an Australian Postgraduate Award.

References

1. Fickett, J. W., and A. G. Hatzigeorgiou, 1997, Eukaryotic Promoter Recognition: Genome Research, v. 7, p. 861-878.
2. Bajic, V. B., S. L. Tan, Y. Suzuki, and S. Sugano, 2004, Promoter prediction analysis on the whole human genome: Nature Biotechnology, v. 22, p. 1467 - 1473.
3. Pedersen, A. G., P. Baldi, Y. Chauvin, and S. Brunak, 1998, DNA Structure in Human RNA Polymerase II Promoters: J. Mol. Biol., v. 281, p. 663-673.
4. Florquin, K., Y. Saeys, S. Degroeve, P. Rouze, and Y. Van de Peer, 2005, Large-scale structural analysis of the core promoter in mammalian and plant genomes: Nucl. Acids Res., v. 33, p. 4255-4264.
5. Fukue, Y., N. Sumida, J.-i. Nishikawa, and T. Ohyama, 2004, Core promoter elements of eukaryotic genes have a highly distinctive mechanical property: Nuc. Acids Res., v. 32, p. 5834-5840.
6. Fukue, Y., N. Sumida, J.-i. Tanase, and T. Ohyama, 2005, A highly distinctive mechanical property found in the majority of human promoters and its transcriptional relevance: Nuc. Acids Res., v. 33, p. 3821-3827.
7. Kanhere, A., and M. Bansal, 2005, Structural properties of promoters: similarities and differences between prokaryotes and eukaryotes: Nucleic Acids Research, v. 33, p. 3165-3175.
8. Choi, C. H., G. Kalosakas, K. Rasmussen, M. Hiromura, A. R. Bishop, and A. Usheva, 2004, DNA dynamically directs its own transcription initiation.: Nucleic Acids Res., v. 32, p. 1584-90.
9. Tsai, L., L. Luo, and Z. Sun, 2002, Sequence-dependent flexibility in promoter sequences.: J. Biomol. Struct. Dyn., v. 20, p. 127-34.
10. Gabrielian, A., D. Landsman, and A. Bolshoy, 1999-2000, Curved DNA in promoter sequences: In Silico Biol., v. 1, p. 183-96.
11. Lisser, S., and H. Margalit, 1994, Determination of common structural features in Escherichia coli promoters by computer analysis: Eur J Biochem, v. 223, p. 823-830.
12. Wang, H., M. Noordeweier, and C. J. Benham, 2004, Stress-Induced DNA Duplex Destabilization (SIDD) in the E. coli Genome: SIDD Sites Are Closely Associated With Promoters: Genome Research, v. 14, p. 1575-1584.

13. Ohler, U., H. Niemann, G. Liao, and G. Rubin, 2001, Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition: Bioinformatics, v. 17, p. S199-206.
14. Baldi, P., Y. Chauvin, S. Brunak, J. G. Anders, and G. Pedersen, 1998, Computational Applications of DNA Structural Scales: Int. Conf. Intell. Syst. Mol. Biol., p. 35-42.
15. Ota, T., Y. Suzuki, T. Nishikawa, T. Otsuki, T. Sugiyama, R. Irie, A. Wakamatsu, K. Hayashi, H. Sato, K. Nagai, K. Kimura, H. Makita, M. Sekine, M. Obayashi, T. Nishi, and T. Shibahara, 2004, Complete sequencing and characterization of 21,243 full-length human cDNAs: Nat Genet., v. 36, p. 40-5.
16. Suzuki, Y., and S. Sugano, 2003, Construction of a full-length enriched and a 5'-end enriched cDNA library using the oligo-capping method: Methods Mol. Biol., v. 221, p. 73-91.
17. Suzuki, Y., R. Yamashita, S. Sugano, and K. Nakai, 2004, DBTSS, DataBase of Transcriptional Start Sites: progress report 2004.: Nucleic Acids Res, v. 32(Database issue), p. D78-81.
18. Ivanov, V. I., and L. E. Minchenkova, 1994, The A-form of DNA: in search of the biological role: Mol Biol (Mosk), v. 28, p. 1258-71.
19. Sivolob, A. V., and S. N. Khrapunov, 1995, Translational positioning of nucleosomes on DNA: the role of sequence-dependent isotropic DNA bending stiffness: J Mol Biol., v. 247, p. 918-31.
20. Blake, R. D., and S. G. Delcourt, 1998, Thermal stability of DNA: Nucleic Acids Res., v. 26, p. 3323-32.
21. Breslauer, K., R. Frank, H. Blocker, and L. Marky, 1986, Predicting DNA duplex stability from the base sequence: Proc. Natl. Acad. Sci. USA, v. 83, p. 3746-50.
22. Satchwell, S. C., H. R. Drew, and A. A. Travers, 1986, Sequence periodicities in chicken nucleosome core DNA: J. Mol. Biol., v. 191, p. 659-75.
23. el Hassan, M., and C. Calladine, 1996, Propeller-twisting of base-pairs and the conformational mobility of dinucleotide steps in DNA: J Mol Biol., v. 259, p. 95-103.
24. Witten, I. H., and E. Frank, 2005, Data Mining: Practical machine learning tools and techniques: San Francisco, Morgan Kaufmann.
25. Platt, J., 1998, Fast Training of Support Vector Machines using Sequential Minimal Optimization, *in* B. Schoelkopf, C. Burges, and A. Smola, eds., Advances in Kernel Methods - Support Vector Learning, MIT Press.
26. Keerthi, S. S., S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, 2001, Improvements to Platt's SMO Algorithm for SVM Classifier Design: Neural Computation, v. 13, p. 637-649.