Improving Promoter Prediction Using Multiple Instance Learning

P. J. Uren, R.M. Cameron-Jones, A.H.J. Sale

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

Abstract. Promoter prediction is a well known, but challenging problem in the field of computational biology. Eukaryotic promoter prediction, an important step in the elucidation of transcriptional control networks and gene finding, is frustrated by the complex nature of promoters themselves. Within this paper we explore a representational scheme that describes promoters based on a variable number of salient binding sites within them. The multiple instance learning paradigm is used to allow these variable length instances to be reasoned about in a supervised learning context. We demonstrate that the procedure performs reasonably on its own, and allows for a significant increase in predictive accuracy when combined with physico-chemical promoter prediction.

1. Introduction and Biological Context

Deoxyribonucelic acid (DNA) stores the instructions for building complex biological organism. Simply speaking, this information is arranged in genes, many of which (but not all) code for proteins — the general functional units of biological systems. DNA is a long polymeric molecule constructed from monomers (of which there are four). Within computational fields, this is commonly represented as a string of letters from the alphabet of A, C, T and G (representing the bases Adenine, Guanine, Cytosine and Thymine); the length unit is base pairs (or *bp*) and is analogous to simple string length. Although DNA is a double helix *in vivo*, it is common to give the sequence of only one strand as the bases bind complementarily; one strand can be inferred from the other. For the purposes of this paper, we will concentrate on the eukaryotic *homo sapiens* genome. Relatively little of the human genome is actually accounted for by genes [1], making the localisation of them highly important. Furthermore, the expression (activity) level of genes is governed by complex regulatory networks involving the binding of proteins called transcription factors to the DNA molecule. This regulatory network ensures the correct temporal, spatial and contextual expression of each gene. A promoter can be thought of as a gene header — a short portion of DNA which is not transcribed, but allows the cellular transcriptional machinery to recognise and bind at the correct location on the DNA molecule. The point within the promoter after which the sequence is transcribed is called the transcription start site (or *TSS*). Transcription factor binding sites are often clustered around, or within, the promoter region [2-5]. Transcription factors (regulatory proteins) bind to these sites, modifying the expression level of the gene. A common method for identifying transcription factor binding sites is via position weight matrices (PWMs). Position weight matrices model the likelihood of a given nucleotide appearing at a given location within a binding site; one matrix is needed for each transcription factor. Given a putative binding site, the model gives the quality of match and a threshold can be used to predict sites. This paper explores a promoter representational scheme based around the location of these transcription factor binding sites.

Prestridge [3] describes an approach based on representing promoters by the location of transcriptional elements. This approach uses a collection of position weight matrices to build a profile of the promoter region. For each transcriptional element considered, the ratio density of occurrences within promoter sequences, as compared to non-promoter sequences, is calculated. A score for a putative promoter is calculated by summing the density scores from the profile which match the transcriptional elements found within the sequence in question. In contrast to the method proposed here however, this does not take into account the relative positioning of sites.

Kondrakhin and colleagues [2] propose a similar technique to that of Prestridge, although using consensus sequences rather than position weight matrices to identify binding sites. The salient difference is that they consider the localisation of binding sites. They split the promoter into regions and construct a two-dimensional matrix representing the occurrence of each binding site within each region. Classification is performed from this matrix via a weighted sum using a threshold.

Other work has also investigated the density of particular oligonucleotides in promoter regions [6, 7]. Within the work of Narang et al. [7] a statistical model is created from a dataset of positive instances without the need for a collection of weight matrices (or other models of the motifs sought). Negative examples or a background genomic model are also not required. The fact that this method does not require pre-existing models of the motifs sought is a major advantage, as many transcription factors either do not have (or have poorly supported) models.

The efficacy of characterising promoters by the distribution of salient motifs within their primary sequence has clearly been established in previous work. The focus here is on a formulation which is amenable to solution within the classic supervised machine learning framework. To this end the next section explores a possible approach utilising the common attribute-vector representation.

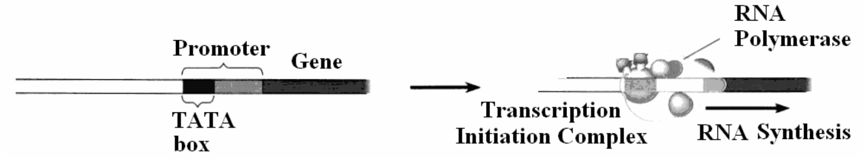

Fig. 1. The process of transcription showing the binding of transcription factors to the DNA molecule within the promoter region (adapted from [8]).

2. Promoter Prediction Using Transcription Factor Binding Sites

A significant number of approaches have previously been presented aimed at solving the promoter prediction problem [9-17]. Bajic [15] et al. report that none of the programs they tested in their review were able to produce a combined sensitivity and positive predictive value of more than 65%, with many falling well below this. We use the same accuracy measures here as used in [15] and [18]. Here we explore a new approach to promoter prediction utilising the existing idea of describing a promoter by the location and name of transcription factor binding sites which fall within it. A relatively simple mechanism is employed for locating these binding sites; the putative promoter region is scanned using position weight matrices (PWMs). This leads to a new consideration: the problem of variable length instances. If one scans the promoter sequence using a weight matrix this produces a series of matches (i.e. there may be zero or more sites that exceed the threshold). Repeating the process searching for binding sites for other distinct transcription factors produces a heterogeneous set of matches. This set is the collection of attributes used to describe a single promoter, or instance, within the dataset. Although it is conceivable that the number of distinct factors for which binding sites are sought is known *a priori*, the number of matches for each matrix is unknown and indeed variable across promoters. This introduces an issue for most established classifier learning schemes, as variable length instances are not supported. It is possible to place an upper bound on the number of matches to a given matrix — one more than the difference between the length of the motif and the length of the sequence being searched. This introduces a new problem though. Such a representation produces instances of extremely high (albeit fixed) dimensionality. With this increased dimensionality, a greater amount of training instances are needed to facilitate the learning of salient concepts from the data. This is not practical as there are a finite and relatively small number of positive instances available. Rather than pursuing this thread of at best marginal utility, the problem can be reformulated using multiple instance learning.

3. Multiple Instance Learning

Within the classic paradigm of supervised machine learning the learner is provided with the correct answers to a set of training instances. Within this paper a variation on this technique called multiple instance learning [19] is used. The fundamental differences are the organisation of instances and the availability of class information. Rather than a single instance being described by a vector of attribute values and a class value, instances are grouped into bags. Instances no longer have classes, but rather it is the bag which has a class value attached. It has been suggested that the multiple instance representation lies in generality somewhere between the attribute-vector representation commonly found in supervised learning and the relational representation associated with the field of inductive logic programming [20]. The multiple-instance problem is really a generalisation of the classic supervised learning problem. Alternatively, for ease of expression, classic supervised learning is a special case of MIL where each bag contains only a single instance.

Several approaches have been devised to construct classifiers for multiple instance learning problems [19, 21-24]. Xu and Frank [25], later followed by Ray and Craven [26], explored logistic regression methods. They propose two methods for determining the bag-level class probability – one based on the arithmetic mean of the instance level probability estimations, the other on the geometric mean. Their underlying generative model does not assume that the bag class is determined by only a single instance; a point of interest within this work given more consideration later. Multiple instance learning has been successfully applied to several domains including drug activity prediction [19, 23] and scene recognition [22, 27, 28] where it deals well with ambiguity with respect to which part of the image is of interest [29]. However, to our knowledge this is the first time it has been utilised for promoter prediction.

4. Multiple Instance Learning for Promoter Prediction

With the above approach, it is possible to modify the promoter representation to avoid the problem of variable length instances. Instead of a promoter mapping to an instance, it will instead be represented by a bag. Individual instances will be positive matches within the promoter sequence for the PWM of a given transcription factor. Note that conceptually an instance (a PWM match) can now be described with a small, fixed number of attributes. These are the name of the transcription factor in question and the location of the hit within the sequence of the promoter (i.e. an index). Each instance is assigned to a bag. The bag represents the promoter on which the matches occurred. That is, each promoter is represented in the dataset by a single bag and each bag contains all the (putative) TFBSs found within that promoter. In fact, the variability has not been eliminated from the dataset, it has simply moved to a higher level – the number of instances within a bag. Table 1 presents a small sample dataset, described in the ARFF format [30]. It has two putative promoters (p1, a negative exemplar and p2, a positive exemplar). The first (p1) is described by two PWM hits and the second (p2) by five PWM hits.

Table 1: A sample MIL promoter dataset in ARFF format

```
@relation MIL_SAMPLE

@attribute PROMOTER_NAME {p1, p2}
@attribute TF_MAT_NAME {MA0001, MA0002, MA0003}
@attribute HIT_LOCATION numeric
@attribute PROMOTER? {yes, no}

@data
p1, MA0001, 36,  no
p1, MA0003, 124, no
p2, MA0001, 12,  yes
p2, MA0001, 34,  yes
p2, MA0002, 56,  yes
p2, MA0003, 89,  yes
p2, MA0003, 156, yes
```

It is common when presenting the application of MIL to a problem domain to also present results obtained from applying a non-MIL classifier to the dataset to demonstrate the improvement apparent from using MIL [19]. This is not meaningful here since the instances are simply TFBSs. Individually they do not represent a promoter and hence a non-MIL classifier could not be expected to learn promoter-level concepts from them. For this reason, standard classifiers are not considered when discussing the result possible from a multiple instance learning solution to this problem.

The above paragraph exposes a caveat about this application of MIL. Strictly speaking, Dietterich et al. [19] specify that a bag is positive if any instances within the bag are positive, or the bag is negative if none of the instances in the bag are positive. However, the problem formulation that was given above does not quite align with this. Here, a bag contains positive instances (TFBS hits which make this promoter bag a positive) and negatives (TFBS hits which do no impart positivity to the bag). So far this matches with the general description of an MIL problem, however here it is not simply the case that a single positive TFBS hit in a bag makes the bag positive. Rather, a more complicated underlying concept exists: some combination of positive instances makes the bag positive. Xu and Frank [25] introduce the idea that the label of a bag is determined from an equal and independent contribution of all the instances within the bag. However, within this application, it is reasonable to assume that there is some dependence between the binding sites discovered.

5. Materials and Methods

We explore the application of MIL promoter prediction within two contexts. In the first, it is applied independently. A dataset is generated using a segment of chromosome 21 from the human genome containing fifty-six known promoters. For each promoter, 150bp upstream and downstream of the TSS are extracted (i.e. 300bp in total). Non-promoter sequences are also extracted from the same region, also of 300bp in length each and totalling 560 instances.

In the second context, MIL is applied as a post-processing step to putative promoters predicted by another promoter prediction methodology — specifically physico-chemical promoter prediction (PCPP) [18]. Here, the full DNA segment is provided to the PCPP layer (note that the PCPP layer is trained on a separate segment of the same chromosome). Two scenarios are considered — passing all instances to the MIL layer or only passing the positive-classified instances. In both cases, the predicted promoter locations (there are 61, 21 correct, 40 incorrect) are taken as TSSs and 300bp windows are extracted around these. When passing all instances, positive instances not correctly identified by the PCPP layer (there are 35) are included as false negatives. True negative instances are generated in the same fashion as above (there are 560). Considering a second level as re-labelling the instances, there are four possible transitions; each type is of interest. The first is a transition from FP to TN (improving accuracy), the second a transition from TP to FN (decreasing accuracy), the third a transition from FN to TP (improving accuracy) and the last being from TN to FP (decreasing accuracy). Table 2 shows the distribution of instances after applying the PCPP layer, but before applying MIL.

Table 2: Distribution of instances before applying MIL

		Actual Class	
		Positive	*Negative*
PCPP Prediction	*Positive*	21	40
	Negative	35	560

Each element of the dataset described above (i.e. each 300bp segment of DNA) is searched for binding sites using 128 position weight matrices from JASPAR [31] — a high-quality, publicly available repository of matrices. The output from this search for binding sites is arranged as per the multiple instance learning paradigm described above. That is to say, fundamentally, each instance is a matrix hit, described by the name of the matrix and an index into the sequence representing where the hit occurred. Instances are assigned to bags representing the promoter from which they were drawn. In addition to a class label, when evaluating the MIL layer as a second step after PCPP, each bag can also be given the prediction made by the physico-chemical promoter prediction software. This allows a multiple-classifier approach utilising the prediction of the lower PCPP layer within the MIL layer. An example dataset (demonstrating the application of just the MIL layer without the PCPP prediction) was presented in Table 2.

As a source of classifiers, MILK [32], a toolkit of multiple instance learning algorithms written as an extension to WEKA [30], is used. Some of the algorithms in MILK are not applicable either due to data incompatibilities or excessive runtime. The logistic regression algorithms presented by Xu and Frank [25] are theoretically well suited to this application, capable of handling the data representation, and have reasonable runtime. Hence, the investigation is concentrated on the use of these. Results are presented for the classifiers MILRARITH (Multiple Instance Logistic Regression with Arithmetic Mean), MILRGEOM (Multiple Instance Logistic Regression with Geometric Mean) and MIRBFNetwork (Multiple Instance Radial Basis Function Network). All experiments are performed using stratified ten-fold cross-validation.

The statistical test employed here is the Wilcoxon signed rank test [33] as described by Conover [34]. Observations are the F-measure for each fold before and after the change in classifier (Note that the split of the dataset into folds is equivalent for all such experiments). Improvements in F-measure, sensitivity or PPV are considered statistically significant if the p-value is less than 0.05.

All of the classifier learning schemes mentioned expose the regression ridge parameter. Empirically it was observed that this parameter influences the quality of classifiers produced. To select a value for this, a nested tenfold cross-validation approach was used. The inner cross validation takes the 90% of the original dataset provided for training in each fold of the outer cross-validation and trains the classifier using a range of possible values for the ridge parameter. Each training of the classifier is itself a complete tenfold cross-validation. The F-measure of each resultant classifier is determined and the parameter value which produces the best F-measure is used to train a final classifier for the given fold of the outer cross validation.

6. Results and Discussion

To put the following results in perspective, one must first consider the performance of the physico-chemical promoter prediction software without the MIL augmentation. Before applying the search for TFBSs and MIL, there are 21 true positives, 35 false negatives, 40 false positives and 560 true negatives. This equates to a sensitivity of 0.38 and a positive predictive value of 0.34. Ideally a balance between sensitivity and positive predictive value is desired. To capture this, here we use F-measure (the harmonic mean of sensitivity and positive predictive value) to represent the quality of the classifier produced.

6.1. PCPP and MIL on Positive-Classified Instances

We begin by examining the main contribution of this paper. That is, the MIL layer utilising only the instances predicted as positive by the PCPP layer. As all the true positives and all the false positives are being provided to the MIL layer, the final count of these is simply the count of true positives and false positives produced by the MIL layer. The final count of true negatives and false negatives is the sum of those produced by the PCPP layer (as none are passed to the MIL layer, they cannot be reclassified) and any new true or false negatives produced by the MIL layer (i.e. false positives correctly reclassified or true positives incorrectly reclassified).

It is not possible for a classifier produced from the positive-predictions-only dataset to demonstrate an improvement in sensitivity over just the PCPP layer. This is obvious if one considers that sensitivity is calculated by dividing the number of true positives by the sum of false negatives and true positives. The denominator of this expression is invariant here since a true positive reclassified becomes a false negative and a false negative reclassified becomes a true positive. The numerator however can be decreased but never increased (it is possible to incorrectly change a true positive to a false negative but there are no false negatives provided to the classifier which might be correctly reclassified to true positives). Hence, the upper bound on sensitivity is that which was produced by the PCPP layer, specifically 0.38.

The results of running the three selected classifier learning schemes using only positively classified instances from the PCPP layer are presented in Table 3. The most striking feature is the improvement apparent from the MIRBFNetwork classifier when propagating only positive-classified instances. The results indicate that the PCPP classifier and the MIRBFNetwork classifier are complementary. The MIRBFNetwork is better at separating true from false positives in the PCPP classifications than the other two classifier types, although all show improvement over just the PCPP layer.

Table 3: Classification performance using PCPP-positive instances. Entries marked in bold show a statistically significant improvement over the base-level PCPP classifier.

	MILRARITH	*MILRGEOM*	*MIRBFNetwork*
Sensitivity	0.37	0.37	0.37
Positive Predictive Value	**0.42**	**0.43**	**0.91**
F-Measure	**0.39**	**0.40**	**0.52**

6.2. MIL promoter prediction performance in isolation.

Having demonstrated that MIL can be utilised to improve the classification performance of the PCPP approach, it is instructive to consider how well it might function on its own. Table 4 shows the performance of the three selected classifier learning methods operating in isolation.

The results in Table 4 show that each of the classifier learning methods utilised was not capable of matching the PCPP layer performance in isolation. This demonstrates the efficacy of including the lower layer. The F-measure of about 0.25, although low, is competitive with a lot of existing promoter prediction approaches. It is however much less than the PCPP layer was capable of in isolation.

Table 4: Isolated MIL layer. Only the F-measure is presented.

	MILRARITH	*MILRGEOM*	*MIRBFNetwork*
F-Measure	0.25	0.25	0.26

6.3. PCPP and MIL on All Instances

Within this section we consider passing all instances from the PCPP layer regardless of whether they were classified as positive or negative. Recall however, that each instance is appended with the PCPP level classification result. The results for running each of the classifiers using this approach are presented in Table 5. Recall that the PCPP layer achieved a sensitivity of 0.34 a PPV of 0.38 and an F-measure of 0.36. Bold entries in Table 5 are significantly better than the PCPP layer performance.

The results are somewhat mixed. Here, MILRGEOM shows a statistically significant improvement in PPV and a corresponding improvement in F-measure. The MIRBFNetwork classifier scores quite poorly on sensitivity and by extension also F-measure due to a large number of negative predictions.

Table 5: Classification performance using all instances. Bold entries signify a statistically significant improvement over the base-level PCPP classifier.

	MILRARITH	*MILRGEOM*	*MIRBFNetwork*
Sensitivity	0.37	0.35	0.15
Positive Predictive Value	**0.40**	**0.44**	**1.00**
F-Measure	0.38	**0.39**	0.26

7. Conclusions and Further Work

Within this paper a multiple instance formulation of the promoter prediction problem was introduced and several algorithms were tested. It has been demonstrated that, with an appropriate selection of learning scheme and parameters, promoters can be predicted using multiple instance learning and a representation based on the location of transcription factor binding sites. Furthermore, the classification performance of this and a physico-chemically based promoter prediction procedure can be improved by arranging the two in a combined classifier. By using a database such as JASPAR, the performance of such an approach can be expected to improve as more experimentally verified binding sites become available.

The most expensive operation is searching raw DNA sequences for binding sites. In general, the application of this approach in isolation is computationally infeasible. However, by considering a multiple classifier system where the MIL layer is only applied to positive-classified instances from the lower PCPP layer, computational requirements can be sufficiently reduced such that the approach becomes practical. Moreover, this multiple classifier system achieves the best performance in terms of F-measure (0.52), improving upon the PCPP layer (which in this scenario achieved an F-measure of 0.36 and in general is capable of an F-measure of approximately 0.40). Putting this in context, the F-measure of 0.52 beats 6 out of the 9 approaches investigated by Bajic et al. [15].

There are potential avenues for reducing the runtime requirements of this MIL approach. Most obviously, one could scan for binding sites using fewer matrices. In line with this, the identification of which matrices produce hits that are used by the classifier would allow biological insight into promoter function. Those matrices which are not important for classification could then be removed allowing for an improvement in runtime performance.

Further to this, there are additional possibilities for combining classifiers. Here we explored only the inclusion of the lower layer prediction as an attribute at the higher layer, but there is extensive research into multiple classifier systems including approaches such as bagging [35], boosting [36] and stacking, which may improve classification performance.

References

1. International Human Genome Sequencing Consortium, *Finishing the euchromatic sequence of the human genome.* Nature, 2004. **431**: p. 931-945.
2. Kondrakhin, Y.V., A.E. Kel, N.A. Kolchanov, A.G. Romashchenko, and L. Milanesi, *Eukaryotic promoter recognition by binding sites for transcription factors.* Comput. Appl. Biosci., 1995. **11**: p. 477-488.
3. Prestridge, D.S., *Predicting Pol II Promoter Sequences using Transcription Factor Binding Sites.* Journal of Molecular Biology, 1995. **249**: p. 923–932.
4. Berman, B.P., Y. Nibu, B.D. Pfeiffer, P. Tomancak, S.E. Celniker, M. Levine, G.M. Rubin, and M.B. Eisen, *Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome.* Proc. Natl. Acad. Sci., 2002. **99**(2): p. 757-762.
5. Frith, M.C., M.C. Li, and Z. Weng, *Cluster-Buster: finding dense clusters of motifs in DNA sequences.* Nuc. Acids Res., 2003. **31**(13): p. 3666-3668.
6. Kel, A.E., N.A. Kolchanov, V.V. Kapitonov, M.P. Ponomarenko, A.E. Likhachev, H.A. Lim, and L. Milanesi. *Computer analysis and recognition of functional sites on the base of oligonucleotide patterns distributions.* in *Second International Conference on Bioinformatics, Supercomputing and Complex Genome Analysis*. 1993. St. Petersburg Beach, Florida, USA.

7. Narang, V., W. Sung, and A. Mittal, *Computational modeling of oligonuceotide positional densities for human promoter prediction.* Artificial Intelligence in Medicine, 2005. **35**(1-2): p. 107-119.
8. Campbell, N.A., L.G. Mitchell, and J.B. Reece, *Biology*. 5th ed. 1999, Menlo Park, CA: Benjamin/Cummings Publ. Co., Inc.
9. Ohler, U., *Promoter Prediction on a Genomic Scale---The Adh Experience.* Genome Res., 2000. **10**(4): p. 539-542.
10. Ohler, U., G.-c. Liao, H. Niemann, and G.M. Rubin, *Computational analysis of core promoters in the Drosophila genome.* Genome Biol., 2002. **3**(12).
11. Ohler, U. and H. Niemann, *Identification and analysis of eukaryotic promoters: recent computational approaches.* Trends Genet., 2001. **17**(2): p. 56-60.
12. Ohler, U., H. Niemann, G. Liao, and G. Rubin, *Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition.* Bioinformatics, 2001. **17**(Suppl 1): p. S199-206.
13. Fickett, J.W. and A.G. Hatzigeorgiou, *Eukaryotic Promoter Recognition.* Genome Research, 1997. **7**: p. 861-878.
14. Abeel, T., Y. Saeys, E. Bonnet, P. Rouze, and Y.V.d. Peer, *Generic eukaryotic core promoter prediction using structural features of DNA.* Genome Res., 2008. **18**(2): p. 310-323.
15. Bajic, V.B., S.L. Tan, Y. Suzuki, and S. Sugano, *Promoter prediction analysis on the whole human genome.* Nature Biotechnology, 2004. **22**: p. 1467 - 1473.
16. Pedersen, A.G., P. Baldi, Y. Chauvin, and S. Brunak, *The biology of eukaryotic promoter prediction-a review.* Computers and Chemistry, 1999. **23**(3-4): p. 191-207.
17. Oppon, J. and W. Hide, *A Statistical Model for Prokaryotic Promoter Prediction.* Genome Informatics, 1998. **9**: p. 271-273.
18. Uren, P., R.M. Cameron-Jones, and A. Sale. *Promoter Prediction Using Physico-chemical Properties of DNA*. in *The 2nd International Symposium on Computational Life Science*. 2006. Cambridge, UK: Springer-Verlag.
19. Dietterich, T.G., R.H. Lathrop, and T. Lozano-Perez, *Solving the Multiple Instance Problem with Axis-Parallel Rectangles.* Artificial Intelligence, 1997. **89**(1-2): p. 31-71.
20. Zucker, J.D. and J.G. Ganascia. *Changes of representation for efficient learning in structural domains*. in *Thirteenth International Conference on Machine Learning*. 1996. Bary, Italy: Morgan Kaufmann.
21. Auer, P. *On learning from multi-instance examples: Empirical evaluation of a theoretical approach*. in *The Fourteenth International Conference on Machine Learning*. 1997: Morgan Kaufmann.
22. Maron, O. and T. Lozano-Perez. *A Framework for Multiple-Instance Learning*. in *Advances in Neural Information Processing Systems*. 1998: MIT Press.
23. Zhang, Q. and S.A. Goldman, *EM-DD: an improved multiple-instance learning technique.* Neural Information Processing Systems, 2001. **14**(10).
24. Zhou, Z.-H. and M.-L. Zhang, *Solving multi-instance problems with classifier ensemble based on constructive clustering.* Knowledge and Information Systems, 2007. **11**(2): p. 155 - 170.

25. Xu, X. and E. Frank. *Logistic regression and boosting for labeled bags of instances*. in *8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*. 2004: Springer-Verlag.
26. Ray, S. and M. Craven. *Supervised versus multiple instance learning: An empirical comparison*. in *The 22nd International Conference on Machine Learning*. 2005. New York: ACM Press.
27. Maron, O. and A.L. Ratan. *Multiple-instance learning for natural scene classification*. in *Fifteenth International Conference on Machine Learning*. 1998. San Francisco: Morgan Kaufmann.
28. Zhou, Z.-H. and M.-L. Zhang. *Multi-Instance Multi-Label Learning with Application to Scene Classification*. in *Advances in Neural Information Processing Systems 19*. 2007: MIT Press.
29. Zhang, Q., S.A. Goldman, W. Yu, and J.E. Fritts. *Content-Based Image Retrieval Using Multiple-Instance Learning*. in *Nineteenth International Conference on Machine Learning*. 2002. Sydney, Australia.
30. Witten, I.H. and E. Frank, *Data Mining: Practical machine learning tools and techniques*. 2nd Edition ed. 2005, San Francisco: Morgan Kaufmann.
31. Sandelin, A., W. Alkema, P. Engstrom, W.W. Wasserman, and B. Lenhard, *JASPAR: an open-access database for eukaryotic transcription factor binding profiles.* Nucl. Acids Res., 2004. **32**(suppl_1): p. D91-94.
32. Xu, X. (2003). *Statistical learning in multiple instance problems.* Unpublished Masters Thesis, University of Waikato.
33. Wilcoxon, F., *Individual Comparisons by Ranking Methods.* Biometrics, 1945. **1**: p. 80-83.
34. Conover, W.J., *Practical nonparametric statistics*. 1980: Wiley.
35. Breiman, L., *Bagging Predictors.* Machine Learning, 1996. **24**(3): p. 123-140.
36. Freund, Y., *Boosting a weak learning algorithm by majority.* Information and Computation, 1995. **121**(2): p. 256–285.