TRAINING SET DESIGN FOR PATTERN DISCOVERY WITH APPLICATIONS TO PROTEIN MOTIF DETECTION

YANLI SUN, ZHENGYUE DENG, GIRI NARASIMHAN
Bioinformatics Research Group (BioRG), School of Computer Science, Florida International University, Miami, FL 33199, USA.

KALAI MATHEE
Department of Biological Sciences, Florida International University, Miami, FL 33199, USA.

Supervised pattern discovery techniques have been successfully used for motif detection. However, this requires the use of an efficient training set. Even in cases where a lot of examples are known, using all the available examples can introduce bias during the training process. In practice, this is done with the help of domain experts. Whenever such expertise is not available, training sets are usually picked at random. We present a new strategy for designing good training sets that uses phylogenetic trees to automatically reduce the bias in training sets. When applied to helix-turn-helix motif detection, we show that the technique improved the error rates over a random choice of training set. More specifically, false positive rates show a marked improvement.

1. Introduction

Motifs are small conserved regions in related proteins that exhibit similar three-dimensional folds and share functional properties. Motif detection is an important facet of protein classification and functional annotation. Motif detection methods have been reviewed extensively and can be classified broadly as consensus-based methods, profile-based methods, and methods based on pattern discovery [1, 10, 19].

Pattern discovery techniques have been successfully used to tackle motif detection problems in protein sequences [2, 3, 7, 12, 14, 16-18]. Both supervised and unsupervised techniques have been developed for this problem [12, 16]. The supervised pattern discovery approach, as used in motif detection, strives to find statistically significant patterns that are present in known examples of the motif and that are known to be absent in sequences that are not examples of the same motif. The compilation of such significant patterns is then used to design a detector for that motif.

One of the main problems related to supervised pattern discovery (and machine learning, in general) is the need for identifying a good training set [11]. Pattern discovery is a learning process and the training set is the very source of

knowledge for the learning program. Thus, the selection of the training set is critical to the success of the learning phase, and consequently, to the success of the detection or classification algorithm.

There are several problems associated with the selection of appropriate training set [11]:

1. Incorrect labeling: This problem can be overcome in motif detection problems by only using those motifs that have been verified by biological experimentation.
2. Under-representation: When certain subtypes are under-represented, it can lead to bias in the training set and in the learning process, and can adversely affect the performance of the detection process. Consequently, the resulting detector may have many false negatives for that motif subtype.
3. Over-representation: If certain subtypes are over-represented, then the training set is likely to get biased or over-trained, resulting in many false positives for those subtypes. An extreme example of bias can occur when all the members of a training set are identical. In this case, the learning process cannot learn anything barring that one example. The resulting classification is likely to be severely limited in its ability to perform detection or classification.

Many possible approaches have been proposed for designing training sets for learning programs. All these approaches strive to achieve a balanced representation of the subtypes in the training set. In general, there are no clear-cut ways to deal with under-representation, because often the subtypes of a class may not be known in advance. Thus there is no way of knowing whether a particular subtype has been excluded from the training set and therefore not "learnt" by the program. In contrast, over-representation can be addressed.

In this paper we show how to design good training sets for motif detection based on pattern discovery. Our approach addresses both under-representation as well as over-representation, and exploits the fact that we are dealing with protein sequences. In simple terms, a good training set must have examples of different subtypes of the motif, and must not have too many examples of motifs that are too similar. The central idea behind our approach is to build a phylogenetic tree based on all known examples of the motif (without worrying about the bias being introduced), and then to carefully pick individuals from the tree to form a good training set.

We compare the outcome of a motif detection program that uses an automatic, phylogeny-based, training set selection algorithm against a motif

detection algorithm that uses a randomly chosen training set. Experimental results show that it is possible to achieve acceptable performance for the motif detection by using a phylogeny-based training set selection algorithm, and that it surpasses the performance of a random choice of training set. In particular, we show that the false positive rate is improved significantly over a method that picks the training set uniformly at random from the training set. While it was not possible to match the performance obtained by using carefully handcrafted training sets, the main advantage of our method is that it obviates the need for an expert biologist to help design the training set.

2. Motif Detection

A motif is a portion of a protein sequence that has a specific structure and is functionally significant. The presence of motifs in a protein is very useful for characterizing and classifying that protein. The helix-turn-helix (HTH) motif was the first protein motif to be discovered for site-specific DNA recognition. The interaction of regulatory proteins with promoter elements in DNA requires the protein to have a HTH motif in order to bind to DNA. The HTH motif is a highly specialized super-secondary structure containing two alpha helices separated by a small turn [4, 8, 13, 15]. The motif is about 22 residues in length. The turn contains about 4 residues and the angle between the two helices is approximately 120° [13]. Although much is known about some of the conserved residues in the HTH motif, what makes HTH motif detection non-trivial is that the motif shows considerable variability in its amino acid composition. In addition, the second helix binds to the DNA molecule in a sequence-specific manner, which means that different HTH motifs bind to different DNA fragments. This in turn implies that the HTH motifs must necessarily show sufficient variation in the second helix so that HTH motifs from different regulatory proteins do not bind to the same DNA fragment, since otherwise they would interact with the same promoter region and regulate the same gene. All these issues make HTH motif an excellent model system to experiment with motif detection methods.

GYM is a HTH motif detection algorithm based on a supervised pattern discovery approach and developed by Narasimhan *et al.* [12]. GYM was successfully trained to detect HTH motifs. Experiments with over 1000 proteins have estimated its false negative rate (i.e., sensitivity) to be about 2% and its false positive rate (i.e., accuracy) to be about 7% [10, 12]. As with any supervised pattern discovery method, GYM works in two phases: a pattern discovery (or pattern mining) phase and a motif detection phase. In the pattern

mining phase, GYM is provided with a training set, which it mines for significant patterns and stores in a pattern dictionary. In the detection phase, GYM examines how well a given protein sequence matches the patterns in the dictionary so as to predict the presence and location of the motif in that protein. GYM was subsequently trained and extended to detect another DNA-binding motif called homeodomain motifs [12].

It is clear is that the performance of a program such as GYM is intimately tied to the quality of the training set used in its pattern discovery phase. In the case of GYM, the training set consisting of 88 HTH motifs was handcrafted by expert biologists for an earlier (profile-based) HTH motif detection algorithm [5, 6]. Another critical component in the pattern discovery phase of GYM was the choice of a "support threshold", which was used to determine the significance of a pattern based on its occurrence in the training set [12]. This threshold represents the trade-off between the detection sensitivity and the detection accuracy. A pattern is said to be significant only if the number of its occurrences exceeds the threshold. The threshold was determined using straightforward permutation experiments, which can easily be extended to detection of a new motif. However, the luxury of an expertly hand-crafted choice of a training set is not possible for a new motif.

Redundancy and bias in the training set are the result of identical or significantly "similar" elements in the motif training set. Two protein motifs may share a common sequence pattern if (a) the pattern contains elements that are critical for constituting the motif, or (b) they share high homology. In other words, even though some things may be evolutionarily conserved, it does not mean that they are critical for constituting the motif. In the former case, it would lead to "good" elements in patterns, while in the latter case, it would result in "corrupt" elements in patterns. If there are many corrupt elements in a pattern, then it can lead to false (spurious) patterns that cause false positives in motif detection. If the corrupt elements extend many good elements in a pattern, then it can lead to false negatives in motif detection since these corrupt elements may prevent a good match from occurring.

This paper explores a training set selection (or refinement) algorithm by reducing the bias in the training set, with the goal of improving the overall performance of a motif detection algorithm such as GYM. In this selection algorithm, *phylogenetic trees* are used as an effective tool to identify similarity among sequences. With the help of phylogenetic trees, one can try to ensure that diverse groups are represented in the training set, while also guaranteeing that not too many representatives from a single subtree are selected. Therefore, by

carefully controlling the similarity present among training sequences, it is possible to effectively reduce the probability of spurious patterns in GYM's data mining and improve its overall performance.

With every new training set GYM's pattern discovery phase will produce a different pattern dictionary. Consequently, this will change the ability of the motif detection phase of GYM to predict a motif. The proposed algorithm for training set selection is evaluated by measuring the performance of the prediction phase of GYM when it uses different training sets output by the training set selection program.

3. Algorithm

Similarity between protein sequences or fragments can be computed as a score after using either a multiple alignment or a set of pairwise alignments. Although alignment scores are sufficient to indicate how well the sequences match one another, they are unable to give further information among those sequences, such as evolutionary information. Phylogenetic trees are useful to identify groups of similar protein sequences. They classify the given sequences into a hierarchy of subtrees. Some of the currently available phylogenetic methods (such as CLUSTAL) also provide a length for each branch that roughly measures the evolutionary distance between the sequences represented by the parent and child nodes [9].

Briefly, our method involves assigning a real valued score to each node in the tree. The scores assigned to the nodes attempt to impose a linear metric on the leaves of the tree. Once the scores have been assigned to each leaf node, the algorithm selects a set of leaves that have representatives from all the major subtrees and are sufficiently "spread out". For the following discussion assume that each branch of the phylogenetic tree has a length (evolutionary distance) associated with it.

In order to achieve this, the score assigned to each node is as follows. The score for the root node is 0. For all other nodes with parent node p, the score is given by:

$$Score(p) + \left(a_i \times \Delta^{\frac{h-\ell}{h} - \frac{1}{k}} \right) \tag{1}$$

where Δ = maximum number of children for any node in the tree, h = total height of the tree, i.e., the total evolutionary distance or length from the root to the farthest leaf, ℓ = total evolutionary distance from root to the parent node, a_i

= index of the node among its siblings, and k = small integer constant. The scores are designed such that they increase exponentially as we move (evolutionarily) farther away from the root. Also, for every node the children and their corresponding subtrees are ordered in an arbitrary manner (say, left to right ordering). The scores are such that they "tend to" increase with the left-to-right ordering of the children. Intuitively, what the numbering scheme guarantees is that if two leaf nodes share a common ancestor that is evolutionarily close enough, then their scores are also close enough. But, as is shown later, if their scores are nearly the same, then the chance of both of them being picked for inclusion in the training set is small.

We now describe our algorithm, referred to as SIEVE. First, the input sequences are used to compute a phylogenetic tree. Algorithm SIEVE then computes the scores for each sequence using the recursive scoring function described above in formula (1). The sequences are then sorted on the basis of their scores. SIEVE then invokes the algorithm PICKLEAVES shown below (Figure 1) to select a subset (of desired size) of the input sequences to be used as a training set for our experiments.

4. Implementation

A "vanilla" implementation of GYM was described in an earlier paper [12]. It used a handcrafted training set in its pattern discovery phase. It was modified to obtain two versions as described below.

- GYM-A uses an automatically generated training set (of varying sizes) generated by the above algorithm SIEVE. All known examples of the motif were used as input for SIEVE.
- GYM-R uses a training set (of varying sizes) chosen uniformly at random from the set of all known examples of the motif.

Both the modifications were only to the input to the pattern discovery phase of GYM. The pattern discovery and the motif detection phases of GYM were left unchanged in GYM-A and GYM-R. See the schematic shown in Figure 2.

Algorithm SELECTLEAVES

Input: Set of candidate sequences, *S*
integer *DesiredSize*
Output: Candidates for training set, *Selected*

Begin
Selected := *S*
Num := sizeOf(*Selected*)
while (Num > *DesiredSize*) **do**
Find pair (n_i,n_{i+1}) of minimum distance
if (i = 1) **then** remove n_i from *Selected*
else if (i = N) **then** remove n_{i+1}
else
Middle := (score[n_{i-1}] + score[n_{i+2}]) / 2
if (score[n_i] is closer to Middle than score[n_{i+1}])
then remove n_i from *Selected*
else remove n_{i+1} from *Selected*
return *Selected*
end SELECTLEAVES

Figure 1: Algorithm to select nodes after they have been assigned scores.

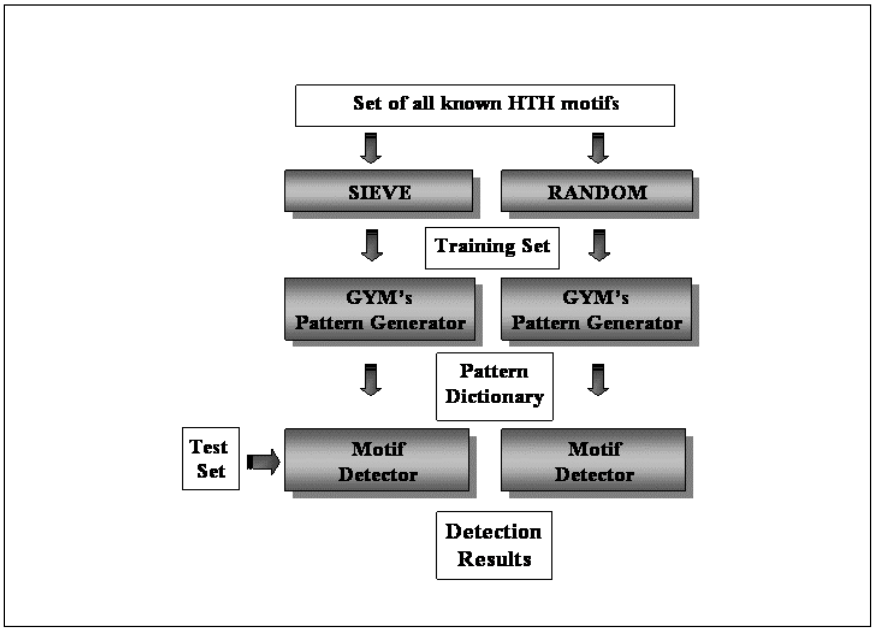

Figure 2: Schematic of the experiments performed

All programs were implemented in C++. The phylogenetic trees used by the SIEVE algorithm were generated using CLUSTALW [9, 20]. The training set output by SIEVE was input to the pattern discovery phase of the GYM-A program. The resulting pattern dictionary was then used by the motif detection phase of GYM, and the performance was evaluated for the number of correct predictions. GYM-R was repeated five times with five different selections of training sets. The performance of GYM-R was thus an average over five runs.

In the scoring function given by formula (1), we used $k = 7$. This number is a measure of the precision to which one wants to compute the scores. The larger the value of the constant k the greater the precision with which that formula would be computed.

One would expect the vanilla version of GYM to have the best performance, since the training set was handcrafted with the help of biologists with expert knowledge. Our experiments were designed to evaluate how GYM-A and GYM-R measure up against the performance of GYM. The subset of results of GYM that could be independently verified by another program was used as a "gold standard" in order to evaluate the correctness of the results of GYM-A and GYM-R.

Feeding the input sequences into CLUSTALW [9, 20] generated the phylogenetic trees, which were then input to the algorithm, SELECT. The training set output by SELECT was input to the pattern discovery phase of the GYM program. The resulting pattern dictionary was then used by the motif detection phase of GYM, and the performance was evaluated for the number of correct predictions.

The results of our experiments are summarized in a graph shown in Figures 2 and 3. Figure 2 shows the false negative rates for our experiments as the number of selected sequences is varied. Figure 3 shows a similar graph with the false positive rates.

It is clear that the SELECT algorithm outperforms the algorithm that picks the training set uniformly at random. With the false negative rates, the difference is marginal. The difference is striking and significant with the false positive rates.

5. Results

Our experiments were designed to evaluate how GYM-A and GYM-R measured up against the performance of GYM. The prior work on GYM, which is based on pattern discovery, and a second method, which was based on the profile method, was useful in generating a list of correct results to evaluate the error

rates of GYM-A and GYM-R. A total of 809 sequences were used in the testing phase, out of which 93 were known to be negative sequences (i.e., do not contain a motif) and 716 were known to have a HTH motif. The false negative and false positive error rates of GYM-A and GYM-R are shown in a graph in Figures 3 and 4 respectively. In both graphs, the performance of the original GYM algorithm is shown as a thick black horizontal line; it clearly outperforms both GYM-A and GYM-R. It is clear that as the number of training sequences increased, the false negative rates tended to decrease, while the false positive rates tended to increase.

A given motif sequence generates a false negative if the system has not been trained using any sequence similar to the one being considered. The probability of this happening decreases as the number of sequences used for training is increased. Figure 3 shows that as the number of sequences used for training was increased from 40 to 100, the false negative error rate for GYM-A dropped from 41% to 17%. GYM-R also showed a similar reduction. GYM-A outperform GYM-R for training set sizes exceeding 50. In fact, for training set sizes above 60, the difference was more than one standard deviation of the GYM-R error rates.

A “negative” sequence i.e., one that does not have a motif, can generate a false positive if it has patterns that it shares with some training sequences, but that are not specific to the motif. The probability of false positive hits increases as the number of sequences used for training is increased. Figure 4 shows that as the number of sequences used for training was increased from 40 to 100, the false positive rate for GYM-A increased from 4% to 56%. At the same time, the false positive rate for GYM-R increased from 9% to 70%. With regard to the false, positive rates, GYM-A clearly outperforms GYM-R, proving that the SIEVE algorithm does reduce the bias in the training set.

6. Discussion and Conclusions

This paper presents a promising approach for training set selection in pattern discovery algorithm with specific applications to motif detection. The proposed method uses information provided by phylogenetic trees to control the amount of similarity among the selected training sequences, thus controlling the amount of duplication of information in the training set. When designing training sets, there is a trade-off between the number of false negative hits and the number of false positive hits. Theoretically, there is no algorithm that can automatically figure out the optimal similarity threshold without further biological knowledge.

This research leads to many interesting questions, especially about the relationships between training sets and phylogenetic trees: Where in the phylogenetic tree are the false negatives and false positives located? If they appear to clump together in the tree, then can increasing or decreasing the representation from that part of the tree decrease the number of false negatives and/or false positives? Is the performance sensitive to the tree construction method used? Is it possible to design algorithms that iteratively add or delete items from the phylogenetic tree with the goal of decreasing the error rate? Are there lessons to be learnt by analyzing the relative positions of the sequences from efficient handcrafted training sets? Is there a correlation between the amount of overlap of the training sets employed by GYM-A or GYM-R and the handcrafted training set employed by GYM?

Acknowledgments

Research of Giri Narasimhan was supported in part by NIH Grant P01 DA15027-01.

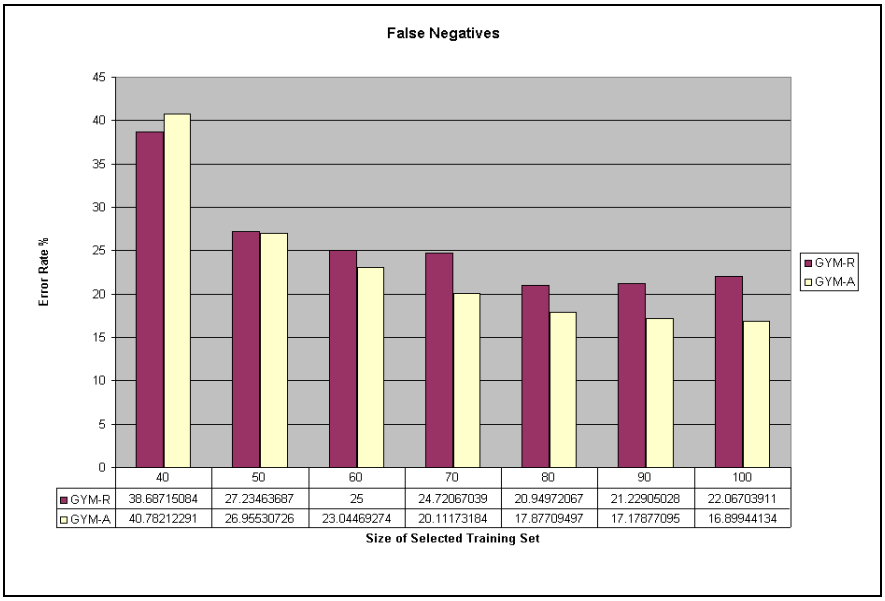

Figure 3: False negative rates for the experiments as the number of selected sequences is varied.

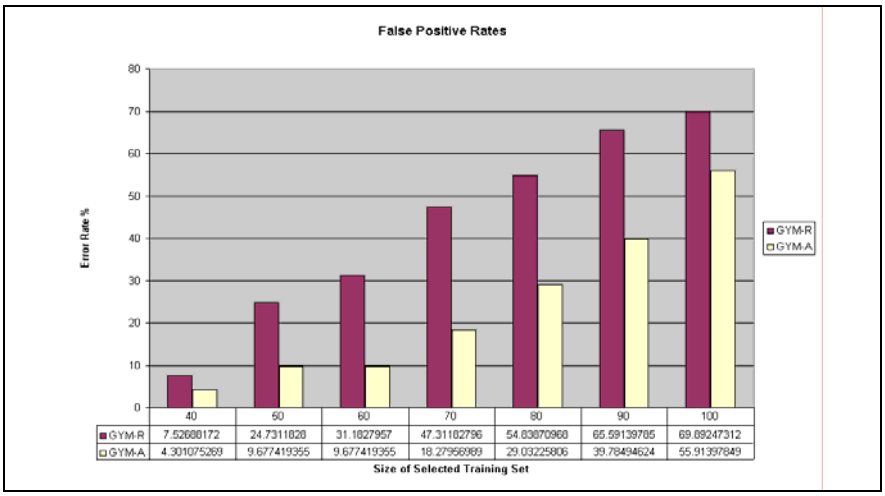

Figure 4: False positive rates for the experiments as the number of selected sequences is varied.

References

1. P. Bork and T.J. Gibson, *Methods Enzymol*, **266**, 162 (1996).
2. A. Brazma, I. Jonassen, E. Ukkonen, and J. Vilo, *Proc Int Conf Intell Syst Mol Biol*, **4**, 34 (1996).
3. A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert, *J Comput Biol*, **5**, 279 (1998).
4. R.G. Brennan and B.W. Matthews, *J Biol Chem*, **264**, 1903 (1989).
5. I.B. Dodd and J.B. Egan, *Protein Eng*, **2**, 174 (1988).
6. I.B. Dodd and J.B. Egan, *Nucleic Acids Res*, **18**, 5019 (1990).
7. Y. Gao, K. Mathee, G. Narasimhan, and X. Wang. *String Processing and Information Retrieval (SPIRE)*. 1999.
8. S.C. Harrison and A.K. Aggarwal, *Annu Rev Biochem*, **59**, 933 (1990).
9. F. Jeanmougin, J.D. Thompson, M. Gouy, D.G. Higgins, and T.J. Gibson, *Trends Biochem Sci*, **23**, 403 (1998).
10. K. Mathee and G. Narasimhan, *Methods Enzymol*, **370**, 250 (2003).
11. T.M. Mitchell, *Machine Leanring*. 1997, McGraw-Hill,
12. G. Narasimhan, C. Bu, Y. Gao, X. Wang, N. Xu, and K. Mathee, *J Comput Biol*, **9**, 707 (2002).
13. H.C. Nelson, *Curr Opin Genet Dev*, **5**, 180 (1995).
14. C.G. Nevill-Manning, T.D. Wu, and D.L. Brutlag, *Proc Natl Acad Sci U S A*, **95**, 5865 (1998).
15. C.O. Pabo and R.T. Sauer, *Annu Rev Biochem*, **61**, 1053 (1992).
16. I. Rigoutsos and A. Floratos. *Proc 4th Annual Intl Conf Comput Mol Biol (RECOMB)*. 1998.
17. I. Rigoutsos and A. Floratos, *Bioinformatics*, **14**, 55 (1998).
18. I. Rigoutsos, A. Floratos, L. Parida, Y. Gao, and D. Platt, *Metab Eng*, **2**, 159 (2000).
19. W.R. Taylor, *Protein Eng*, **2**, 77 (1988).
20. J.D. Thompson, D.G. Higgins, and T.J. Gibson, *Nucleic Acids Res*, **22**, 4673 (1994).