Motif Detection in Protein Sequences

(Extended Abstract)

Yuan Gao * Kalai Mathee † Giri Narasimhan ‡§ Xuning Wang ¶

April 16, 1999

Abstract

We use methods from Data Mining and Knowledge Discovery to design an algorithm for detecting motifs in protein sequences. Based on this approach, we have implemented a program called "GYM". The *Helix-Turn-Helix Motif* was used as a model system on which to test our program. The program was also extended to detect *Homeodomain motifs*. The detection results for the two motifs compare favorably with existing programs. In addition, the GYM program provides a lot of useful information about a given protein sequence.

1 Introduction

A *Motif* is a region or portion of a protein sequence that has a specific structure and is functionally significant. Protein families are often characterized by one or more such motifs. Detection of motifs in proteins is an important problem since motifs carry out and regulate various functions, and the presence of specific motifs may help classify a protein.

We describe a new approach to the problem of automatic motif detection. We use methods from Data Mining and Knowledge Discovery to design an algorithm that displays increased sensitivity as compared to existing algorithms, while maintaining good accuracy and also providing additional information about a given protein sequence. Unlike previous approaches, our algorithm does not use statistical methods. However, our algorithm does need a "training set" of aligned sample motifs. The basic assumption is that specific combinations (of which there could be many) of residues in specific locations within the motif are responsible for imparting the structure and the functionality to the motif. With this in mind, our algorithm searches for patterns from the sample training set that are present in a new protein sequence. Our approach has similarities to an independently developed method designed by Rigoutsos and Floratos [RF98], which does arbitrary motif detection on unaligned protein sequences. This paper describes how our algorithm can be implemented efficiently. The resulting program is called GYM. Finally, we describe our experiments with the *Helix-Turn-Helix Motif*, which was used as a model system on which to test our program. Results of tests on *Homeodomain motifs* are also reported.

2 Motifs in Protein Sequences

As mentioned above, motifs share a common structure and function. We describe the structural and functional properties of the two motifs used in this study.

*yuangao@us.ibm.com; IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

†kmathee@utmem2.utmem.edu; Department of Microbiology and Immunology, University of Tennessee – Memphis, 858 Madison Avenue, Memphis, TN 38163.

‡Corresponding Author; giri@msci.memphis.edu; Department of Mathematical Sciences, The University of Memphis, Memphis, TN 38152; Fax: (901) 678-2480.

§Partially funded by NSF grant CCR-940-9752 and by a grant from Cadence Design Systems, Inc.

¶Xuning.Wang@wl.com; Parke-Davis Pharmaceutical Research, 2800 Plymouth Road, Ann Arbor, MI 48105.

Helix-turn-helix Motifs The helix-turn-helix motif was the first protein motif to be discovered for site-specific DNA recognition. This motif has been widely investigated and there exists substantial knowledge of the chemical interactions of specific residues. Crystal structures of many of the proteins containing these motifs are also available. Limited information is also available from mutational analysis of the motif and the effect of specific amino-acid substitutions on motif structure. This motif is common to many DNA-binding proteins and plays a crucial role in their binding to DNA. Thus studying these motifs provides an excellent model system for the study of protein motifs that are used in site-specific recognition.

Features of the helix-turn-helix motif have been reviewed extensively by Pabo and Sauer [PS92] and Nelson [Nel95]. Briefly, it consists of two α-helical structures separated by a turn. The motif is about 20 to 22 residues in length. The turn consists of three to four amino acids, and the two helices make an angle of approximately 120° [Nel95]. One of the two helices is responsible for binding to DNA in a sequence-specific manner, and is referred to as the "recognition helix". In most proteins, the second of the two helices is the "recognition helix". Residues of the recognition helix interact directly with bases in the major groove of the DNA. A combination of residues in both the helices are believed to be responsible for maintaining the appropriate angle between the two helices. Proteins with helix-turn-helix motifs share only limited sequence homology in the motif region; the dissimilarity is attributed to the sequence-specific interactions with the bases in the DNA. Most proteins have at most one helix-turn-helix motif; however, we will discuss a family of proteins that have more than one such motif. All the properties mentioned above make automatic recognition of helix-turn-helix motifs an interesting (and non-trivial) algorithmic problem.

Homeodomain Motifs Proteins containing the homeodomain motif play an important role in plant and animal development. The homeodomain [STH89] motif is made up of three α-helices and an extended N-terminal arm. The first and second α-helices pack against each other in an anti-parallel arrangement, while the third α-helix lies perpendicular to them. The third helix is the *recognition helix*; like its counterpart in the helix-turn-helix motif, it interacts with DNA in the major groove and provides the DNA-binding specificity. However, unlike the helix-turn-helix unit, the 60-residue homeodomain forms an independent folded structure and can independently bind to DNA. It is interesting to note that the homeodomain motif contains a canonical helix-turn-helix structure. Mutational and evolutionary analyses and crystal structures of these domains are also available. The homeodomain motifs were chosen for testing because they are almost thrice as long as the helix-turn-helix motifs.

3 Motif Detection

3.1 Existing Methods

To aid in the detection of motifs in protein sequences, classical methods involved carefully crafting a "consensus" sequence to reflect highly conserved residues in the motif. Pabo and Sauer [PS92] constructed a consensus sequence for helix-turn-helix motifs based on a multiple alignment of known motif sequences. One simple method to detect helix-turn-helix motifs is to look for the occurrence of such a consensus sequence. More general "consensus" sequences are also possible (as the ones maintained by PROSITE [Bai92]). Inspired by the concept of regular expressions, such generalized consensus sequences consist of a sequence of sets of amino acids where amino acids within the same set could substitute each other in that position. Nevill-Manning et al. [NMWB98] presented a method for discovering motifs from families of aligned protein sequences. Their method was based on automatically constructing such generalized consensus sequences. Earlier, Wu and Brutlag [WB96] had showed a way of constructing substitution sets in a statistically significant manner. Another feature of the algorithm due to

Nevill-Manning et al. [NMWB98] involved constructing a set of such consensus sequences with the assumption that any one of the sequences in the set could describe the motif. Additionally, the software based on their method (EMOTIF) provided the user with parameters to tradeoff sensitivity to specificity. These parameters are claimed to control the number of false positives.

Other sophisticated detection schemes are all statistically motivated. These are typified by the *Profile* method described by Gribskov et al. [GLE90]. The first step, once again, involved making a multiple alignment of known motifs. The next step typically involved computing a probability matrix or a *Score Matrix*, which assigns a different score to each possible residue at each position in the motif. Intuitively, the entries of this matrix represent a measure of the probability that a certain residue occurs in that location, normalized by the background frequencies for that residue. Minor variants exist in the methods employed to compute the position-specific scoring matrix as well as in scoring a match (see, for example, [Gus97]). Given a score matrix, the detector, when given an input protein sequence, computes a weighted score for every subsequence of the input sequence, and reports the subsequence with the highest score as the detected motif, as long as this score is above a certain threshold.

Dodd and Egan [DE90] showed how to compute such a matrix in a simple manner. The probability values they used were simply the frequency of a residue normalized against the background frequencies for that residue. Other statistically-based methods for detecting motifs include that of using a Hidden Markov Model [GBEB97] and Gibbs Sampling [LAB+93].

3.2 Motif Detection – New Method

Our method is not based on statistical methods. It discovers patterns in known motifs to compute a pattern dictionary. Detection of a motif in a new protein sequence is then a function of which patterns from the dictionary are present in the new protein sequence.

The first assumption that our algorithm makes is that an appropriate length of the motif is known beforehand. This is a reasonable assumption to make since this is true for most of the known motifs (as in the case of helix-turn-helix motifs and homeodomain motifs). For example, there is ample evidence to show that a helix-turn-helix motif lies within a window of size at most 22. The second assumption is that a reasonably large number of motifs are known and have been detected and verified by experiment in the standard way. The training set can be chosen from these known motifs. The third assumption is that a **combination** of key residues are sufficient to constitute the necessary physical structure and to give it the functionality; the rest of the parts of the motif may serve other purposes. Note that this is where we differ from the assumptions made for the other methods. While many of the methods attach separate significance to the occurrence of specific residues in specific locations in the motif, they do not account for the reinforcing effect of a combination of specific residues. For instance, residue x in location l_x may be very significant only if residue y is in location l_y and residue z in location l_z simultaneously. Residue x in location l_x may not be very significant otherwise and may not occur frequently in that location in known motifs and thus may not have a high score in the score matrix. It is likely that the patterns in the pattern dictionary discovered by our method represent such "reinforcing" combinations, helping in the detection of new motifs. Finally, we assume that a "good" combination of residues must occur "frequently enough" to be called a valid pattern for the motif. To account for relatively rare reinforcing combinations, we opted for setting an absolute threshold value to decide whether a combination occurs "frequently enough", as opposed to a requirement that a combination occurs in a specified percentage of the sequences in the training set.

Rigoutsos and Floratos [RF98] independently devised a method to discover unknown motifs without doing alignment, i.e., the training set for their program is a set of unaligned protein sequences. Their method is based on similar ideas of generating patterns, which in turn could be used to perform detection. Other related methods are reviewed by Brāzma et al. [BJEG95]. While the overall philosophy of these methods coincide with ours, our algorithm differs from them in that it detects known motifs after being trained on a set of aligned sequences for the same motif, thus making use of all available knowledge about the motif.

Our methods also share some overlap with that of Nevill-Manning et al. [NMWB98]. The fundamental differences lies in the way the threshold (this concept is explained below in Section 4.1) is used; they require that their motifs "cover" some percentage of the sequences in the training set.

4 The New Algorithm

Here we present our new algorithm for detecting known motifs in protein sequences. We will refer to this algorithm as the "Pattern Dictionary" method. Based on this new algorithm, we have implemented a program (called "GYM") that detects helix-turn-helix motifs. This program was also modified and retrained to detect homeodomain motifs in protein sequences. Note that the program can be modified to detect other motifs.

The algorithm requires that an approximate length of the motif be known beforehand and that a reasonably large number of motifs are known and have been detected and verified by experiment in the standard way. The training set can be chosen from these known motifs. The algorithm consists of two parts. The first part is a preprocessing step that needs to be performed only once. The second part is where the actual motif detection takes place.

The preprocessing phase can be called the **Pattern Mining** phase. The input to this phase is the set of known and aligned motifs, or the *Master Set*. The output is a *Pattern Dictionary* consisting of frequently occurring *Patterns* within the Master Set. The preprocessing phase is described in detail in Section 4.1. The input to the second part, or the **Detection** phase, consists of the pattern dictionary output from the preprocessing step and the input protein for which the motif detection needs to be performed. The detection phase is described in Section 4.2.

The output of the detection algorithm will indicate whether or not the protein sequence contains a motif, the location of this motif, a score indicating the confidence of the prediction, along with a list of proteins from the master set that share high sequence homology with the detected motif as inferred from matching patterns from the dictionary.

4.1 Preprocessing: "Pattern Mining"

The input to this phase is a master set of aligned motifs without spaces. Thus when two motifs are aligned, either the amino acids in a certain location in the motif match or have a mismatch. Figure 1 shows an example of a set of aligned motif sequences, where each motif is of length 7; this is a hypothetical motif that is simply used to illustrate the method. Note that each of these motif sequences occur at different locations in different proteins as indicated in Figure 1.

Location in Seq.	Sequence 1 2 3 4 5 6 7	Protein Name
14	`G V S A S A V`	Ka RbtR
32	`G V S E M T I`	Ec DeoR
33	`G V S P G T I`	Ec RpoD
76	`G A G I A T I`	Ec TrpR
178	`G C S R E T V`	Ec CAP
205	`C L S P S R L`	Ec AraC
210	`C L S P S R L`	St AraC

Figure 1: Aligned Motifs – An example

Every amino acid in each of the motif sequences above is associated with a position in the motif. Thus protein *Ka RbtR* has amino acid `G` in location 1, `V` in location 2, `S` in location 3, and so on. We thus represent the motif by a sequence of pairs, where each pair <*aa, pos*> consists of an amino acid and its position in the motif. We simplify the notation and denote these amino acids by pairs of symbols such as `G1`, `V2`, and `S3`, respectively. *Ka RbtR* would thus be denoted by the set of pairs {`G1, V2, S3, A4, S5, A6, V7`}.

A **Pattern** is simply a set of pairs. Thus, {`G1, S3, T6`} and {`C1, P4, S5, L7`} are two examples of patterns. Protein *Ec DeoR* contains the pattern {`G1, S3, T6`}, but does not contain the pattern {`C1, P4, S5, L7`}. The *length* of a pattern is defined as the number of pairs in it. Thus, {`G1, S3, T6`} is a pattern of length 3; this pattern is also shared by protein *Ec CAP* and *Ec RpoD*. The *support* of a pattern is the number of proteins in which it appears. For the 7 motif sequences in Figure 1, the patterns {`G1, S3, T6`} and {`G1, S3, V2`} have a support of 3, while pattern {`G1, S3`} has a support of 4. A pattern is called a *significant* pattern (or a frequent pattern), if its support is no less than a certain *threshold*. A significant pattern is called *maximal* if it is not contained in any other significant pattern. For a threshold value of 3, the pattern {`G1, S3, T6`} is significant, but not maximal, since the maximal significant pattern {`G1, I7, S3, T6, V2`} contains it. The pattern mining phase outputs a list of all maximal frequent patterns. This list will henceforth be referred to as the *pattern dictionary*. For a threshold value of 3, the dictionary that will be output for the example in Figure 1 would be that in Figure 2. If we lower the threshold to 2, the dictionary that will be output would be as shown in Figure 3.

Pattern length	Maximal Patterns	Support
2	{ `S3, P4`}	3
2	{ `S3, S5`}	3
3	{ `G1, T6, I7`}	3
3	{ `G1, S3, T6`}	3
3	{ `G1, V2, S3`}	3

Figure 2: Dictionary of frequent patterns for the aligned motifs from Fig. 1 (THRESHOLD = 2).

The algorithm goes through at most m (length of the motif) iterations. In the i-th iteration, it generates all frequent patterns of length i. In the i-th iteration, the algorithm first generates a collection of potentially frequent patterns and then their supports are computed to verify if they are frequent enough. The collection of potential patterns generated in the i-th iteration consists only of those patterns that are obtained by the set union of two frequent patterns of length $i-1$ that differ in exactly one item. Even this observation is not enough to efficiently generate all potentially frequent patterns. We first present the algorithm in Figure 4 followed by a discussion of the implementation details that make it efficient.

Pattern length	Maximal Patterns	Support
2	{ `S3, P4` }	3
3	{ `G1, S3, V7`}	2
5	{ `G1, V2, S3, T6, I7` }	2
7	{ `C1, L2, S3, P4, S5, R6, L7` }	2

Figure 3: Dictionary of frequent patterns for the aligned motifs from Fig. 1 (THRESHOLD = 2).

Algorithm *Pattern-Mining*
Input : Motif length m, support threshold T, and list of aligned motifs.
Output : Dictionary L of frequent patterns.
1. Generate all frequent patterns of length 1 and insert into list L_1.
2. **for** $i = 2$ to m **do**
3. **for** every pair of patterns $p, q \in L_{i-1}$ such that $|p \cap q| = i - 1$ **do**
4. Insert pattern $p \cup q$ into list E_i
5. **for** every pattern $p \in E_i$ **do**
6. **if** $(support(p) > T)$ **then**
7. Insert p into L_i.
8. **if** $(|L_i| \leq 1)$ **then**
9. **return** $L = \cup_i L_i$

Figure 4: Pattern Mining Algorithm

Implementation Details Enumerative algorithms that generate such a dictionary of frequent patterns are likely to be inefficient because of the combinatorial explosion in the number of possible patterns. A simple counting argument shows that if the *motif length* is denoted by m, there are $20m$ possible patterns of length one (because there are only 20 amino acids), and $O((20m)^k)$ possible patterns of length k. However, it was recently shown [Par99] that the number of frequent or significant patterns is also bounded by the length of the input, i.e., it is bounded by $O(mn)$, where n is the number of motif sequences in the input.

The naive algorithm of generating all possible patterns of length i and checking whether it is significant or not will clearly be very inefficient. Our algorithm is based on an algorithm from the Data Mining by Agrawal et al. [AMS+96], and is able to avoid the generation of most infrequent patterns by using an efficient screening process to be described below. The basic idea behind the algorithm by Agrawal et al. is that if a pattern occurs frequently, then every subset of this pattern must necessarily be frequent. This also implies that if a pattern does not occur frequently enough, then all supersets of this pattern may be immediately discarded.

In order to make this process efficient, all patterns are stored in a canonical form with their items in sorted order. For example, the canonical form for a pattern {`G1, A2, I4, A5`} would be the sequence {`A2, A5, G1, I4`}. The sorting is done in a simple lexicographic manner. Furthermore, the list of patterns are also sorted (again, in a lexicographic order). Once the dictionary is put in this form, a potentially frequent pattern is generated if two frequent patterns of length $i-1$ share the first $i-2$ items in common. But then, since the list is in sorted order, such patterns are going to be next (or at least close) to each other in the list. To be more precise, a block of k patterns that share the first $i-2$ items are going to contribute $O(k^2)$ potentially frequent patterns of length i. Another important observation that is not obvious is that the list does not need to be sorted if things are processed in a proper order. When a new pattern is generated, it is automatically put at the end of the list. Because of the order in which things are considered, the list remains sorted. For the first iteration, the input sequences are scanned and for every amino acid in the input sequences, either a new pattern of length one is created or the support of an existing pattern is incremented. The implementation resulting from the algorithm is very fast in practice.

4.2 The Detection Algorithm

The detection algorithm is now quite straightforward. It takes as input a motif length m, the dictionary of significant patterns L output by the *Pattern-Mining* algorithm (Figure 4), an integer k representing the number of best matches required as output, and the given protein sequence P to be examined for the motif. We slide a window of length m across the input sequence P. The subsequence of P that lies in the window is then matched against every significant pattern in L. This is performed in a subroutine called Match. Match returns a *Match-Score* that quantifies how well the window matched against the patterns in the dictionary L. While it is convenient to think of Match-Score as a number, it is in reality a collection of measures that describe the quality of the match. The parameters that define the Match-Score is explained later. The k best Match-Scores along with the corresponding window locations is maintained. Finally, all matches from the list of the k best matches whose quality exceeds a pre-specified threshold are reported as possible motif locations. The algorithm is described in Figure 5.

Match Parameters The following are the parameters that define the quality of a match of a window of size m with a dictionary of proteins: **LPM** – the length of the longest pattern matched, **NDP** – the number of distinct positions from the window that matched some pattern, and **NPM** – the number of distinct maximal patterns matched.

Comparing two matches The selection of the list of parameters and the process of comparing two matches has been developed and fine-tuned after a close study of experimental data.

Algorithm *Motif-Detection*

Input : Motif length m, threshold score T, dictionary L of patterns, number of best matches k, and input protein sequence $P[1..n]$.

Output : Information about motif(s) detected.

```
Best-Match-Score = Match(P[1..m], L)
for i = 2 to n - m + 1 do
    Match-Score = Match(P[i..i + m - 1], L)
    Update list of k best matches found so far.
for i = 1 to k do
    if i-th best match-score >= T then
        Report it as possible motif location
```

Figure 5: Motif Detection Algorithm

A significant pattern from the dictionary represents a combination of amino acids in specific locations that (potentially) positively reinforce the motif structure. Thus, the longer the pattern, the greater the number of positive reinforcements to the structure, and consequently, the better the quality of the match. By a similar argument, NDP and NPM are also significant.

Output of Algorithm The algorithm is designed to output the k locations with the highest match score, as long as the scores are above a pre-specified threshold. The algorithm will output the location of the motif as well as the residues in the motif. It also prints the match parameters, i.e., LPM, NDP, and NPM, for the k best matches. The output includes the set of patterns that are matched at the predicted motif location. It also indicates which particular residues in the motif are present in the patterns matched.

Lastly, the output gives a list of proteins from the training set that exhibit the same patterns found in the motif of the input protein sequence. For new protein sequences, this could provide clues to the family to which this protein may belong in terms of its function. This information would become more valuable when it is combined with similar information from other motifs found in the same protein sequence. We also conjecture that a more careful study of the list of patterns matched in a protein could help in determining evolutionary relationships between proteins.

5 Results and Discussion

The pattern dictionary algorithm described above was implemented, trained, and tested.

We first discuss the choice of the training set, also referred to as the "Master Set". In general, the choice of a training set is a non-trivial problem, and could determine the success or failure of a motif detection method. Also, automatic generation of a training set is a difficult problem. We initially used the same training set (91 proteins) as Dodd and Egan [DE90], but subsequently eliminated three proteins. For these three proteins the GYM and DE programs reported different motif locations, and experimental evidence defining the precise locations was not available in the literature. We deleted these proteins so that their motifs could not bias the training set. The three proteins deleted were: (a) SpoOA *Bacillus subtillis* (Assn. No: 134739): GYM predicted positions 5 and 219, DE predicted 198; (b) XylR *Bacillus subtillis* (Assn. No: 98448): GYM predicted position 361, DE predicted 29; (c) pSC101 rep (Assn. No: 281929): GYM predicted a marginal motif at location 103, while DE failed to predict any motif.

Next we discuss the choice of the support threshold T in *Pattern-Mining* algorithm. This parameter represents a tradeoff between the sensitivity and the number of false positives that can be generated by the detection algorithm. When the threshold is higher, while fewer patterns are generated, the resulting patterns are likely to have higher statistical significance. On the other hand, if the threshold is lower, sensitivity is higher since good patterns with lower statistical significance can be detected. The threshold value used for our experiments was equal to the maximum value that optimized the detection of motifs from the training set itself, i.e., if the threshold is any higher, then the detection algorithm failed on some instances from the Master Set itself. For the given training set, we chose a threshold of 4. Once again, automatic choice of the threshold value seems to be a difficult problem.

Since we wanted to test the performance of the GYM program on a diverse set of proteins, the GYM program was then tested on several families of proteins. Some of the sub-families (such as the SigE sub-family) are not represented in the training set. The sequences were down-loaded from GenBank Protein Sequence Database maintained by the National Center for Biotechnology Information (NCBI). For the helix-turn-helix motif, we ran the GYM program on 675 protein sequences. For our tests, we chose to output only the two (i.e., $k = 2$) best locations; choosing $k = 1$ was inadequate because there are a number of proteins with more than one helix-turn-helix motif. We ran two independent sets of experiments on the 675 proteins; one set was run with the assumption that the helix-turn-helix motif was located within a window of 20 residues, while another set was run with a window size of 22. Since there were only minor differences in the two sets, we only present the results for a window size of 22. On the whole, the results for a window size of 20 were subsumed by the results for a window size of 22, i.e., sometimes the results for a window size of 22 may contain a prediction that is missed by the results for a window size of 20.

Of the 675 sequences, GenBank had the motif location annotated for 241. However, the database had no information on how these locations were determined (What program was used to determine the motif location? Were any laboratory experiments performed to verify the claim?). Thus we consider these annotations unverified, as are the predictions of our program. In order to confirm our results with an independent program, we also implemented the score matrix method described by Dodd and Egan [DE90]. For the sake of convenience, we refer to this program as *DE*, and we refer to our program as *GYM*. Among the 675 proteins selected, 93 are proteins involved in metabolic pathways and other enzymatic reactions. These are presumed not have a helix-turn-helix motif, since they are unlikely to bind to DNA. We refer to this family as the "Negates" family (see section 5.3 for details).

The programs GYM and DE disagreed on 68 (approximately 10%) of the sequences. Of these disagreements 23 were from the negates set, indicating a large number of false positives for our program. Another 23 of them were from the "Sigma" family (sigma factor proteins). As discussed in section 5.2, there is evidence to support the helix-turn-helix predictions made by our program for these 23 sigma factor sequences.

It is interesting to note that the DE program, by virtue of its design, makes a sharper distinction between its first choice and its second choice in terms of the weighted scores. This is not true of the GYM program. While this was an asset in dealing with proteins that had two helix-turn-helix motifs, it could be considered a drawback in other cases.

The results are discussed in detail in Sections 5.2- 5.4. The scores are first summarized in Figure 6. The first column specifies the family of proteins tested. The next two columns state the number of sequences tested and the number on which the two programs agreed on a motif location. The two programs are said to agree if they have a common location within their top two choices and if the corresponding scores are above the threshold. The next two columns indicate how many of the sequences tested had published annotations for the motif location and how many of these matched with GYM's predictions. Once again, an annotation is matched if one of the top two locations (if above the threshold) are the same. While this was not known before testing, we noticed that of the 675 proteins analyzed, only 459 of the detected motifs were unique, i.e., 216 of the sequences had motifs that were identical to the motifs in some other sequences. Even if these are discounted, the overall results show that out of 459 ($= 675 - 216$) sequences tested, the two programs agree on 414 ($= 630 - 216$) of them, giving a percentage of about 90%.

We also modified the GYM program to detect homeodomain motifs. An appropriately modified version of the DE program was also used. After training the two programs with 121 sequences, we ran the two programs on 524 protein sequences. There was overwhelming agreement between the two programs. The results are summarized below in Figure 6.

5.1 Master Set

The GYM and DE programs were first tested on the 88 sequences from the Master Set (the same set they were trained with). The two programs agreed on the locations of the motif in all of them.

5.2 Sigma Family

The 304 proteins selected for this set are all sigma factors, which are known to be DNA-binding proteins. The sigma subunit of eubacterial RNA polymerase is required for recognition of promoter sequences and initiation of transcription from those sites [LGG92]. Two major subfamilies of the sigma family of proteins have been identified: (i) the σ^{70} or RpoD subfamily, which is used by most of the "housekeeping" genes expressed during exponential growth, and (ii) the alternative sigma factor subfamily, including RpoS, RpoE, FliA, etc., which are involved in coordinated expression of sets of genes during a change in metabolic or developmental state.

The results for this set of proteins were quite interesting. GYM and DE predictions did not match for 34 of the 304 sequences tested. On closer inspection we found that the predicted locations for 23 of them were about 90-93 residues apart, and that most of them were from the RpoS subfamily. Members of the sigma family are known to have two helix-turn-helix motifs [LGG92, GB97] in regions 3.1 and 4.2, which are about 90-93 residues apart. For the RpoS subfamily, the GYM program detected the motif in region 3.1, while DE picked the one in region 4.2, thereby accounting for the disagreements. It is also interesting to note that the motifs in region 3.1 were not represented in the master set but GYM was still able to detect them.

5.3 Negates Family

Among the 675 proteins analyzed, 93 were specifically chosen as proteins involved in metabolic pathways and other enzymatic reactions. We presumed that these proteins would be unlikely to have a DNA-binding function, and consequently would be unlikely to have a helix-turn-helix motif. We refer to this set of proteins as the "negates" family.

Out of these 93 proteins, GYM predicted a potential helix-turn-helix motif for 23. We initially interpreted these results as "False Positives"; however, inspection of the crystal structure of one of these proteins (Adenylosuccinate Synthetase from *Escherichia coli*, Assn. No: 1942847) revealed three α-helices at locations 183-191, 193-201, and 204-214. The GYM program predicted a motif at locations 188-219, which includes the last two α-helices. This demonstrates that the GYM program is able to detect structural motifs with high sensitivity, and that a helix-turn-helix structure can occur without guaranteeing a DNA-binding function for the motif region.

5.4 LysR and AraC families

The proteins from the LysR family [Sch93] are predominantly similar-sized, autoregulatory transcriptional regulators. In response to different inducers, members of this family of proteins activate divergent transcription of linked target genes or unlinked regulons encoding extremely diverse functions. Mutational studies and amino acid sequence similarities have identified a DNA-binding domain employing a helix-turn-helix motif (residues 1-65).

The AraC/XylS [GSB$^+$97] family of transcriptional regulators includes proteins and predicted polypeptides derived from translation of DNA sequences. Members of this family are about 300 amino acids long and have three main regulatory functions: carbon metabolism, stress response, and pathogenesis.The conserved region contains all the elements required to bind DNA target sequences and to activate transcription from cognate promoters.

The GYM predictions agreed with those from

Motif	Protein Family	How Many Tested	GYM = DE Agree	How Many Annotated	GYM = Annotated
Helix-Turn-Helix Motif (Window Size = 22)	Master	88	88 (100%)	13	13
	Sigma	304	270+23 (96%)	96	82
	Negates	93	70 (75%)	0	0
	LysR	127	125 (98%)	95	93
	AraC	63	54 (86%)	37	29
	Total	675	607+23 (93%)	241	217 (90%)
Homeodomain Motif (Window Size = 60)	Master	121	121 (100%)	121	121
	Rest	403	390 (97%)	385	370
	Total	524	511 (98%)	506	491 (97%)

Figure 6: Summary of Motif Detection Results

DE much more for the LysR family than the AraC family. Among the disagreements in the LysR family was one (BlaA Sc; Assn. No: 461627) for which the published location matches GYM's strong prediction for a motif at location 17; whereas, neither of the top two DE predictions included this location.

Among the disagreements in the AraC family, there were several that were displaced by about 60 residues. This raises the question whether there is a second helix-turn-helix structure in that location (as observed in the Sigma family). There is evidence to suggest that this region containing the second motif may have some biological significance [GSB+97]. It is also interesting to note that the average "score" given by both programs was relatively low for members of the AraC family, suggesting somewhat different characteristics for the helix-turn-helix motifs in this family?

5.5 Homeodomain Motif

Both GYM and DE programs were retrained with homeodomain motifs (60 residues in length) from 121 proteins to generate a corresponding pattern dictionary. The two programs agreed with each other's predictions and with the database annotations approximately 97% of the time. Such a high percentage of agreement may arise because homeodomain motifs are found in closely related proteins, or because the proportion of amino acids conferring DNA-binding specificity (amino acids that are different) is much smaller than the proportion conferring α-helical structure (amino acids that are similar). We are unable to offer any explanations for the cases where the predictions of the two programs differed.

6 Conclusions

The GYM program has excellent ability to predict helix-turn-helix motifs. It appears to have increased sensitivity over the DE program and can detect motifs with greater differences from the training set. On the negative side, the GYM program appears to have a higher number of false positives; this may result from detection of closely placed α-helices that are not directly involved in DNA binding. Modification of the GYM program to detect the longer homeodomain motifs was also successful and resulted in very high agreement with DE and the database sequence annotations.

The previous statistical methods for motif detection are limited in that the sample training set must contain sufficient representation of amino acid substitutions that preserve or reinforce the particular structure and function of the motif. Some combinations of residues in specific locations can reproduce that structure; whereas others cannot. Until we can model these complex molecular interactions, the next best thing will be to detect and enumerate successful combinations of residues that form such motifs.

Acknowledgments

We thank Mu Yang and Xingqiang Wang, who helped in a previous implementation of this algorithm. We thank Yi De for valuable discussions. We are very grateful to Dr. Martha Howe for a careful reading of the paper, for many critical comments, and for help with interpreting our results.

References

[AMS+96] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. MIT Press, 1996.

[Bai92] A. Bairoch. PROSITE: a dictionary of sites and patterns in proteins. *Nucleic Acids Research*, 20:2013–18, 1992.

[BJEG95] A. Brāzma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. Technical Report 113, Department of Informatics, University of Bergen, Norway, Dec 1995.

[DE90] I. B. Dodd and J. B. Egan. Improved detection of helix-turn-helix DNA-binding motifs in protein sequences. *Nucleic Acids Research*, 18(17):5019–26, 1990.

[GB97] T. M. Gruber and D. A. Bryant. Molecular systematic studies of eubacteria, using σ^{70}-type sigma factors of group 1 and group 2. *J. Bacteriol.*, 179(5):1734–1747, 1997.

[GBEB97] W. N. Grundy, T. L. Bailey, C. P. Elkan, and M. E. Baker. Hidden markov model analysis of motifs in steroid dehydrogenases and their homologs. *Biochem. Biophys. Res. Comm.*, 231:760–766, 1997.

[GLE90] M. Gribskov, R. Lüthy, and D. Eisenberg. Profile anlysis. In R. F. Doolittle, editor, *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences*, volume 183 of *Methods in Enzymology*, pages 146–159. Academic Press, 1990.

[GSB+97] M. Gallegos, R. Schleif, A. Bairoch, K. Hofmann, and J. Ramos. AraC/XylS family of transcriptional regulators. *Microbiol. Mol. Biol. Rev.*, 61(4):393–410, 1997.

[Gus97] D. Gusfield. *Algorithm on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

[LAB+93] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 8 Oct. 1993.

[LGG92] M. Lonetto, M. Gribskov, and C. A. Gross. The σ^{70} family: Sequence conservation and evolutionary relationships. *J. Bacteriol.*, 174(12):3843–3849, 1992.

[Nel95] H. C. M. Nelson. Structure and function of DNA-binding proteins. *Current Opinion in Genetics and Development*, 5:180–189, 1995.

[NMWB98] C. G. Nevill-Manning, T. D. Wu, and D. L. Brutlag. Highly-specific protein sequence motifs for genome analysis. *Proc. Natl. Acad. Sci. USA*, 95:5865–5871, May 1998.

[Par99] L. Parida. Pattern discovery in data: Linear bound on irredundant motifs and a polynomial algorithm. Manuscript, 1999.

[PS92] C. O. Pabo and R. T. Sauer. Transcriptional factors: Structural families and principle of DNA recognition. *Ann. Rev. Biochem.*, 61:1053–95, 1992.

[RF98] I. Rigoutsos and A. Floratos. Motif discovery in biological sequences without alignment or enumeration. In *Proc. of the Second Annual International Conference on Computational Molecular Biology, RECOMB 98*, pages 221–227, 1998.

[Sch93] M. A. Schell. Molecular biology of the LysR family of transcriptional regulators. *Annu Rev Microbiol*, 47:597–626, 1993.

[STH89] M. P. Scott, J. W. Tamkun, and G. W. Hartzell. The structure and function of the homeodomain. *Biochim Biophys Acta*, 989(1):25–48, Jul 1989.

[WB96] T. D. Wu and D. L. Brutlag. Discovering empirically conserved amino acid substitution groups in databases of protein families. In *Porceedings of the Fourth International Conference on Computational Biology (ISMB-96)*, pages 230–240, 1996.