Local Prediction Approach for Protein Classification Using Probabilistic Suffix Trees *

Zhaohui Sun **Jitender S. Deogun**

Department of Computer Science and Engineering
University of Nebraska – Lincoln,
Lincoln, NE 68588-0115, USA,
Email: zsun{deogun}@cse.unl.edu

Abstract

Probabilistic suffix tree (PST) is a stochastic model that uses a suffix tree as an index structure to store conditional probabilities associated with subsequences. PST has been successfully used to model and predict protein families following global approach. Their approach takes into account the entire sequence, and thus is not suitable for partially conserved families. We develop two variants of PST for local prediction: *multiple-domain prediction* and *best-domain prediction*. The multiple-domain method predicts the probability that a protein belongs to a family based on one or more significant conserved regions, while the best-domain method does it based on the most conserved region in the query sequence. The time complexity of both of our approaches is the same as that of the global prediction, that is, $O(Lm)$ where L is the depth bound of the tree and m is the size of the query sequence. We tested our algorithms on the Pfam database of protein families and compared the results with the global prediction method . The experimental results show that our approaches have higher accuracy of prediction than that of global approach. We also show that, our local prediction approach is an effective way to extract motifs/domains. Our approaches employ a linear time method for building PST by adapting the linear time construction of Probabilistic Automata reported by A.Apostolico et al.

1 Introduction

Classification of proteins and extraction of motifs have become an active research area in recent years. A variety of approaches such as Pfam (Bateman et al. 1999), PROSITE (Hofmann et al. 1999), SAM (Hughey & Krogh 1998), Gibbs sampling (Thijs et al. 2001, Marchal et al. 2003), MEME (Bailey & Elkan 1995), and Stochastic Dictionary Model (Gupta & Liu 2003) have been developed. These databases are very useful in the analysis of newly discovered protein sequences. By classifying a protein into a family, we can infer its functions based on the known information about the family.

Recently, Probabilistic Suffix trees (PST) approach has been introduced to classify protein families and detect conserved motifs from unaligned sequences (Bejerano & Yona 2000, Ron et al. 1996). This approach is based on a certain feature, "short memory", which is common to most biological sequences. Before PST there were efforts to model sequences with this feature using Markov chains of order L (the memory length of the model) and HMMs. However, both methods have critical limitations for practical use. The size of order L Markov chains grows exponentially with respect to their order, and thus only very low order Markov chains can be applied in practice. The HMMs based method suffers from known learnability hardness results (Gillman & Sipser 1994). Although the PST approach is based on the same observation, it can model rich sources with high efficiency using a reasonable amount of memory. The probabilistic suffix tree has been demonstrated to be a powerful, efficient, and simple-to-use approach to classify protein sequences (Bejerano & Yona 2000). G.Benjerano and G.Yona have applied this approach to classify protein families in Pfam database and showed that the PST model has higher accuracy of prediction than pairwise methods such as Gapped-BLAST, and is as accurate as a HMM trained with a multiple alignment, but much more efficient (Bejerano & Yona 2000). As a global prediction approach, their method takes into account the entire sequence, and thus may not be suitable for families conserved only partially.

In this paper, we develop two variants of PST for performing local prediction. The experimental results show that our approaches have higher accuracy of prediction than that of the global approach.

2 Probabilistic suffix trees

Probabilistic suffix tree, a stochastic model using a suffix tree as the index structure, is employed to serve as a compact representation to organize conditional probabilities distribution for a cluster of sequences. This approach is based on so-called "short memory" feature of natural sequences. That is, the empirical probability distribution of the next symbol given the preceding segment can be accurately approximated by observing no more than the last L symbols in that segment. For example, $P(s_{i+1}|s_0 \cdots s_i) = P(s_{i+1}|s_{i-L} \cdots s_i)$, $i > L$. Each node of a PST is associated with a probability vector that stores the probability distribution for the next symbol given the label of the node as the preceding segment. That is, the probability vector associated with the node θ stores each $P(s_i|label(\theta))$ which represents the probabilities of observing symbol $s_i \in \Sigma$ given that $label(\theta)$

*This research was supported in part by NSF EPSCOR Grant No. EPS-0091900 and NSF Digital Government Grant No. EIA-0091530.

is the preceding subsequence.

One difference of PST from the ordinary suffix tree is that a PST is constructed with reversed sequences. This is to facilitate the process of locating the longest suffix. For more details on the concept of PST, we refer readers to the example of PST given by G.Bejerano and G.Yona (Bejerano & Yona 2000).

A.Apostolico and G.Bejerano introduced a linear time construction of Probabilistic Automata which contains all information stored in the corresponding PST (Apostolico & Bejerano 2000). We adopted their algorithm to construct our PST. This method constructs a PST in $O(n|\Sigma|)$ time (Apostolico & Bejerano 2000).

With a PST we can estimate the probability of each letter by scanning the tree in search of the longest suffix. The probability of the entire sequence is estimated by multiplying the probabilities of each letter from left to right. For example, the probability of a sequence $abracadabra$ with a certain PST can be computed as follows:

$$\begin{aligned} & P(abracadabra) \\ = \; & P(a)P(b|\underline{a})P(r|ab)P(a|ab\underline{r}) \\ & P(c|\underline{abra}) \cdots P(a|abracadab\underline{r}) \\ = \; & \gamma_{root}(a)\gamma_a(b)\gamma_{root}(r)\gamma_r(a)\gamma_{bra}(c)\cdots\gamma_r(a) \end{aligned}$$

where $\gamma_{s'}(\theta)$ is the probability of observing the symbol θ given s' is the preceding segment, and the underlined subsequences represent the longest suffices that appear in the tree (Bejerano & Yona 2000).

To predict if a sequence belongs to a family, we need to compare the probability that the sequence is generated by the PST, with the probability that the sequence is generated by a random uniform distribution.

To compare the two probabilities, the similarity scores are defined as follows:

For a sequence $S =$ "$s_1 \cdots s_n$",

$$\begin{aligned} sim(s_i) &= \frac{P_{PST}(s_i)}{P_{random}(s_i)} \\ sim(S) &= \frac{P_{PST}(S)}{P_{random}(S)} \\ &= \frac{\prod_{1\le i\le n} P_{PST}(s_i|s_1\cdots s_{i-1})}{\prod_{1\le i\le n} P_{random}(s_i)} \\ &= \prod_{1\le i\le n} sim(s_i|s_1\cdots s_{i-1}) \end{aligned}$$

The higher the $sim(S)$ score is, the more confident we are in classifying the sequence into the family. The time complexity of the global prediction is $O(Lm)$ where L is the depth bound of the PST and m is the length of the query sequence, because retrieving each conditional probability requires traversing a path from the root down a tree of maximal depth L, and we need to retrieve m probabilities to compute the overall probability.

3 Local Prediction versus Global Prediction

The prediction process shown in section 2 follows a global prediction approach. G. Bejerano and G.Yona have used this method to model and predict protein families (Bejerano & Yona 2000). One limitation of their method is that as a method of global prediction, it does not perform well for those sequences which are similar to other family members along only a relatively small part of sequences. The local similarities of these sequences may be masked by the long unrelated regions, and thus the probabilities of such sequences can be nearly as low as that of totally unrelated sequences. For these sequences, methods based on local prediction are expected to perform better. G.Bejerano and G.Yona also tried a variant to accommodate local prediction. However, this variant was actually a heuristic that introduces more parameters that must be optimized for each family (Bejerano & Yona 2000).

We developed two variants of PST for performing local prediction: *multiple-domain prediction* and *best-domain prediction*. the multiple-domain method predicts the probability that a protein belongs to a family based on one or more significant conserved regions, while the best-domain method does it based on the most conserved region in the query sequence. The multiple-domain method is an efficient and effective approach for classifying not only the families with one specific domain or motif, but also the families with several separated motifs or domains. The best-domain method is better for detecting the boundary of the most conserved domain, and therefore can be used to extract motifs from sequences.

4 Algorithms of Local Prediction

4.1 Best-domain prediction

If a partial sequence (a domain) is conserved over the sequences of a family, the probability of seeing a symbol at a certain site depends only on the subsequences within the domain, rather than the subsequences outside the domain. Suppose in the sequence "abr<u>acadabra</u>", there is a domain starting from the fourth position (marked with underline). In this case, the probability of "c" in the fifth position is more likely to be $P(c|a)$ rather than $P(c|abra)$. Therefore, for local prediction, we do not compute the similarity score for a whole sequence. Instead, we consider every subsequence staring from different positions. For a sequence $s_1 s_2 \cdots s_m$, to perform a local prediction, we need to consider all the subsequences $s_i \cdots s_j$ starting from s_i $(1 \le i \le m)$ and ending at $s_j (i \le j \le m)$. We calculate the similarity score for each subsequence $S = s_i \cdots s_j$, $1 \le i \le j \le m$:

$$\begin{aligned} & sim(s_i \cdots s_{j-1}s_j) \\ = \; & \frac{P_{PST}(s_i \cdots s_{j-1}s_j)}{P_{random}(s_i \cdots s_{j-1}s_j)} \\ = \; & \frac{P_{PST}(s_i)P_{PST}(s_{i+1}|s_i)\cdots P_{PST}(s_j|s_i\cdot\cdot s_{j-1})}{P_{random}(s_i)P_{random}(s_{i+1})\cdots P_{random}(s_j)} \end{aligned}$$

Therefore,

$$\begin{aligned} & sim(s_i \cdots s_{j-1}s_j) \\ = \; & sim(s_i \cdots s_{j-1}) \times \frac{P_{PST}(s_j|s_i\cdot\cdot s_{j-1})}{P_{random}(s_j)} \end{aligned}$$

We compute each score $sim(s_i \cdots s_j)$ $1 \le i \le j \le m$ using the above relation, and choose the highest score as the score for the local similarity. For example, if the highest score is yielded at the position 30 from the subsequence $s_{10} \cdots s_m$, then the most conserved region is

$s_{10} \cdots s_{30}$, and $sim(s_{10} \cdots s_{30})$ is the score of the local similarity for the sequence.

Since we need to compute scores for subsequences starting from n different sites, it may seem that the time complexity would be larger than the global prediction which only requires one scan of the sequence. However, we devise a way to control the complexity. To keep the time complexity under control, we scan several subsequences simultaneously. When we scan to a certain position i in the sequence, we retrieve all the probabilities $P(s_i|s_j \cdots s_m)$ for all subsequence $s_j \cdots s_m$ $1 \leq j \leq i$ in only one traversal from the root to the leaf labeled with the preceding segment.

A traversal from the root to some leaf in PST retrieves all probabilities $P(s_i|s_j \cdots s_{i-1})$ $i - L + 1 \leq j \leq i$. For the subsequence starting from j $(i - L + 1 < j < i)$, the probability of s_i is $P(s_i|s_j \cdots s_{i-1})$; for $j \leq i - L + 1$, the probability of s_i is $P(s_i|s_{i-L+1} \cdots s_{i-1})$; for $j > i$, the probability of s_i is not needed. Therefore, one traversal from the root to a leaf retrieves all the probabilities that are needed for computing the probabilities for a certain position for all subsequences. Figure 1 shows an example how this works. To compute the scores for all subsequences of a sequence of length m, we only need m traversals in PST, which is the same number we need for the global prediction. Each traversal visits at most L nodes and edges. Therefore, the time complexity of the Best-domain local prediction algorithm is $O(Lm)$ where L is the depth bound of the PST and m is the length of the query sequence, which is the same as that of the global prediction.

To prevent the score of random probability being much larger than 1 , we introduce a parameter d $(d > 1)$. The similarity score is then calculated as follows:

$$sim(s_i) = \frac{P_{PST}(s_i)}{P_{random}(s_i) \times d}$$

The pseudocode for the Best-domain local prediction algorithm is given below: Let $retrieve - probabilities(i)$ denote the procedure that traverses a PST and retrieves the probabilities of position i for all subsequences and let array $probs$ store the probabilities returned by the procedure. $probs[j] = P(s_i|s_j \cdots s_{i-1})$, $i - L + 1 \leq j \leq i$. Let m denote the length of the query sequence.

Algorithm Best-domain:

- $best\_previous \leftarrow$ a large negative number
- $optimal \leftarrow$ a large negative number
- for i from 1 to m
  - $probs[L] \leftarrow retrieve - probabilities(i)$
  - $width \leftarrow \min[L, i + 1]$
  - for j from 0 to $width$
    - $sim(s_{i-j} \cdots s_i) \leftarrow sim(s_{i-j} \cdots s_{i-1}) \times \frac{probs[j]}{P_{random}(s_i) \times d}$
    - $optimal \leftarrow \max[optimal, sim(s_{i-j} \cdots s_i)]$
  - if $i \geq L$
    - $best\_previous \leftarrow best\_previous \times probs[L - 1]$
    - $optimal \leftarrow \max[optimal, best\_prevous]$
  - if $i \geq L - 1$
    - $best\_previous \leftarrow \max[sim(s_{i-L+1} \cdots s_i), best\_previous]$
- return $optimal$

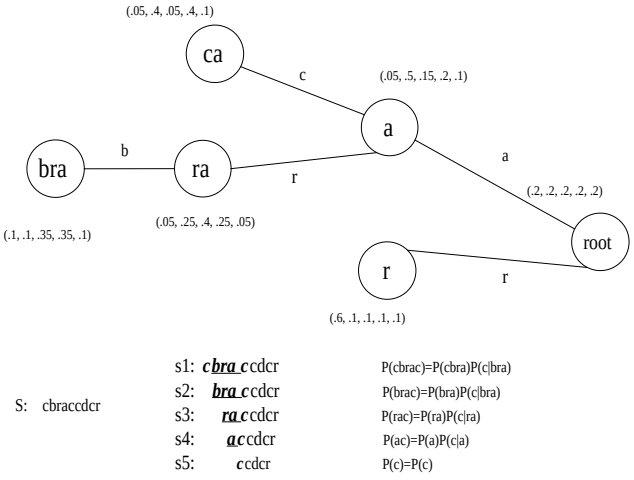

Figure 1: Illustration of computation of a local prediction. Given the PST and sequence S, compute the local scores for subsequences $s1$ - $s5$. The probability of observing c in $s[5]$ in each subsequence is $P(c|bra), P(c|bra), P(c|ra), P(c|a)$, and $P(c)$, respectively. The probabilities can be retrieved by one traversal from the root to the leaf bra.

The algorithm returns the maximum score among all the subsequences. The time complexity of this algorithm is $O(Lm)$ where L is the depth bound of the PST and m is the length of the query sequence.

4.2 Multiple-domain local prediction algorithm

A sequence may have multiple conserved domains. In this case, when we do prediction on the sequence, we want to consider all the conserved domains. We will sum up the similarity scores of separated regions and find the optimal sum. This is very similar to the "repeated" matches alignment algorithm described in Durbin's (Durbin et al. 1998), and can be implemented using Dynamic Programming. We move along a sequence from left to right, computing the highest score for each position. For a sequence $S = s_1 s_2 s_3 \cdots s_{j-L} s_{j-L+1} \cdots s_i \cdots s_j \cdots s_m$, let $max\_before(j)$ denote the best score for all the subsequences $s_i \cdots s_j$ $(i < j - L)$, and $start\_before(j)$ store the value of i which yields the best score; $max\_here(j)$ denote the best score we get for position j, and $max\_so\_far(j)$ denote the best score we get among positions 1 to j.

$$\max\_before(j) = \max_{i<j-L}[sim(s_i \cdots s_j)];$$

$$max\_here(j) = \max_{1 \leq i \leq j-1}[sim(s_i \cdots s_{j-1} s_j) max\_so\_far(i-1)]$$

$$= \max \begin{cases} \max_{[} sim(s_i \cdots s_{j-1}) \times \frac{P(s_j|s_i \cdots s_{j-1})}{P_{random}(s_j) \times d} \\ \times max\_so\_far(i-1)], j - L \leq i \leq j-1 \\ \\ max\_before(j-1) \times \frac{P(s_j|s_{j-L} \cdots s_{j-1})}{P_{random}(s_j) \times d} \\ \times max\_so\_far(start\_before(j)-1) \end{cases}$$

$$max\_so\_far(j) = \max \begin{cases} max\_so\_far(j-1) \\ max\_here(j) - T \\ 0 \end{cases}$$

where $sim(s_i \cdots s_j), 1 \leq i \leq j \leq m$ can be calculated using the same procedure that the Best-domain algorithm uses. To avoid getting local segments with too small score we set a threshold T. Thus, only the motifs whose scores are higher than T are considered by the algorithm.

After we finish a scan of the entire sequence from 1 to m, $max\_so\_far(m)$ is the optimal score of the local prediction for the whole sequence (consists of one or more domains) If we record the parameters in the algorithm, we can trace back to discover the boundaries of the domains found by the algorithm.

5 Results and Discussion

5.1 Protein classification

The best-domain and multiple-domain algorithms were implemented in C++. In our implementation, we used logarithm of all probabilities to avoid overflow or underflow. We tested both algorithms on the protein families found in the Pfam database release 10.0. For our tests we selected 80 families among the large number of families in the database. Each family was divided into a training set and a test set in ratio 2:1. After constructing a PST from the training set of a family, we used the PST to predict the membership of each sequence in the test set and in the SWISSPROT database. We applied the "equivalence number" criterion to determine the threshold of the total score for deciding the membership of a protein (Pearson 1995). This criterion sets the threshold to a value such that the number of false positives (the number of non-member proteins with scores above the threshold) equals the number of false negatives (the number of member proteins with scores below the threshold). The accurate prediction is referred to as the case that the score of a member protein is higher than the threshold and thus is predicted correctly.

To make comparison between the local prediction approaches and the global prediction approach, we also tested the global prediction method on the same data sets using the same criterion. When doing global prediction, we normalized the score of each sequence by its length so that the bias caused by the length difference was eliminated.

Table 1 shows part of the results of all the three algorithms for the 80 protein families. The average error rate of each of the local prediction approaches is much lower than that of the global prediction. The average error rate of the global prediction is 17.6%, while the average error rate of the best-domain and multiple-domain local prediction is 10.9% and 10.2%, respectively. The average error rate of the best-domain and the multiple-domain method is reduced by 38.1% and 42.1%, respectively, compared with the global prediction method. For most protein families, in a sequence the conserved regions only cover part of the sequence. For this reason, the local prediction methods tend to have better overall performance than the global prediction method. For some protein families, the local prediction methods perform significantly better than the global prediction method. For example, for the family A2M_N, the error rate of global prediction is 32%, while the multiple-domain prediction has only 2% error rate. G.Bejerano and G.Yona also tried a local-prediction variant of PST. However, their variant is a heuristic which needs to optimize several parameters for each family. Compared to their method, our local prediction methods are convenient-to-use, since they do not require optimizing parameters for each family.

G.Bejerano and G.Yona compared the PST approach with other protein classification approaches in their previous work (Bejerano & Yona 2000). Their results show that in terms of accuracy of prediction, the PST (in global mode) is much better than a typical BLAST search, and almost as good as an HMM trained from a multiple alignment of sequences. In terms of running time, the PST approach is much faster than the HMM approach. The performance of the PST approach is improved further with our local prediction methods.

5.2 Identify domains/motifs

The local prediction approaches are able to find the most conserved domains in sequences and detect the boundaries of the domains. Therefore, these approaches can be used to identify functionally or structurally important domains/motifs in unknown sequences. We tested the ability of PST in doing this by applying the PST trained from the training data set of the ABC_membrane family to identify the transmembrane domains in five protein sequences. The ABC_membrane family is a very broad protein family and the proteins in the family have a conserved transmembrane domain. The training data set of the ABC_membrane family was from the Pfam database. The test sequences we chose are the full length sequences of the ABC transporter protein of *Chlamydomonas* (Im & Grossman 2001), the *Salmonella* ABC-type transport system protein, the *Homo Sapiens* ABC transporter subfamily B member 4 isoform B, the *Rattus Rorvegicus* ABC transporter 7 protein, and the *Homo Sapiens* ABC transporter subfamily C member 1 isoform 2. All of the proteins have the trans-membrane domain (Table 2). The domain regions predicted by our approach, which are shown in Table 2, match the real domain regions very well. For the *Homo Sapiens* ABC transporter subfamily C member 1 isoform 2 we also identified another ABC transmembrane domain in region 912 - 1181. The region 932 - 1472 of this protein is a ABC-type multidrug transport system and may span multiple domains. Our result indicates that there is a ABC type transmembrane domain in this transporter region. One of the advantages of our approach is that, it is convenient to use our method to identify domains/motifs, because our method computes the boundaries of the conserved domains directly.

5.3 Complexity and running time

Let the total length of the training set be n, the depth bound of PST be L, the alphabet be Σ, and the length of a generic query sequence be m. Building a PST using G.Bejerano and G.Yona's method takes $O(Ln^2)$

Family	Size	Error rate of global prediction	Error rate of best-domain	Error rate of multiple-prediction
14-3-3	214	0.014	0.000	0.000
7tm_4	267	0.326	0.094	0.142
Acetyltransf	2275	0.108	0.216	0.242
2_5_ligase	63	0.302	0.286	0.175
7tm_5	311	0.309	0.058	0.100
acid_phosphat_B	64	0.281	0.156	0.156
2-Hacid_DH_C	494	0.034	0.004	0.028
7tm_6	83	0.313	0.313	0.301
acid_phosphat	181	0.309	0.011	0.105
2-Hacid_DH	413	0.213	0.177	0.220
A1pp	239	0.251	0.063	0.167
A2M_N	106	0.321	0.038	0.019
actin	1716	0.013	0.008	0.013
2-oxoacid_dh	296	0.162	0.000	0.010
A2M	150	0.260	0.013	0.020
A_deaminase	111	0.270	0.198	0.261
2-ph_phosp	18	0.278	0.278	0.167
A_deamin	82	0.061	0.000	0.000
3_5_exonuclease	207	0.324	0.314	0.314
AAA_div	46	0.130	0.000	0.000
adh_short	3193	0.028	0.015	0.022
3-alpha	20	0.250	0.250	0.250
aakinase	621	0.296	0.081	0.077
ADH_zinc_N	1927	0.092	0.026	0.040
3A	94	0.011	0.000	0.011
AAL_decarboxy	37	0.243	0.027	0.108
3Beta_HSD	93	0.312	0.108	0.161
alpha-amylase	1215	0.056	0.002	0.002
A_amylase_inhib	9	0.000	0.222	0.000
aminotran_4	199	0.276	0.010	0.035
3HCDH_N	237	0.114	0.008	0.013
aa_permeases	930	0.204	0.104	0.098
ank	9689	0.002	0.016	0.087
3HCDH	292	0.041	0.010	0.038
arf	247	0.024	0.000	0.000
3H	27	0.259	0.222	0.185
Aa_trans	286	0.329	0.315	0.308
asp	512	0.184	0.023	0.016
AVERAGE	-	0.176	0.109	0.102

Table 1: Comparison of the performance of the local prediction approaches with that of the global prediction approach. The first column gives the names of the families abbreviated as in the Pfam database. The second column gives the numbers of the proteins in the families that are included in the Pfam database. The third, forth, and fifth columns give the error rate of prediction of the global prediction, the best-domain local prediction, and the multiple-domain local prediction, respectively. In our implementation of the three algorithms, we use logarithm of all probabilities. The PSTs were constructed using the parameters $L = 20$, $P_{min} = 0.0001$, and $r = 1.05$. The multiple-domain and best-domain prediction algorithms used the parameter set $T = 7.0$ and $d = 1.5$.

Protein	The region of the transmembrane domain	The domain region predicted by PST
Salmonella ABC-type transport system	15 - 279	14 - 282
Chlamydomonas ABC transporter	440 - 734	457 - 744
Homo Sapiens ABC transporter subfamily B member 4 isoform B	84-180	56-178
Rattus Rorvegicus ABC transporter 7 protein	9 - 205	10-202
Homo Sapiens ABC transporter subfamily C member 1 isoform 2	325 - 596	324 - 595

Table 2: The results of detecting domains/motifs using the local prediction approaches of PST

time, while the construction algorithm used by us takes $O(|\Sigma|n)$. The time complexity of the local prediction algorithms is $O(Lm)$, the same as that of the global prediction algorithm.

G.Bejerano and G.Yona reported that their method for building PST runs for half an hour in average on a Pentium II 300 Mhz PC (Bejerano & Yona 2000). To make a comparison, we compiled and ran the linear time PST-building algorithm on the same machine. For a typical size of protein family (total length = 100000), the linear time algorithm runs for less than 10 seconds.

The global prediction method takes about 5 minutes to predict all sequence in the SWISSPROT database on a Solaris machine. The local prediction methods require more time than the global method. The best-domain method and the multiple-domain method takes about 30 minutes and 40 minutes, respectively, to predict all sequences in the database. Given that the database contains a large number of sequences, and that HMMs take much longer to finish that, the running time of the two local prediction methods is reasonable.

6 Conclusions

The probabilistic suffix tree has been shown to be a powerful and efficient approach for protein classification. The PST captures a common feature of natural sequences called "short memory", and can model rich sources with high efficiency. Building a PST does not require multiple alignment of sequences, and can be done in linear time. In previous work, the PST model following a global prediction approach has been demonstrated to perform well. We introduce two variants of PST for local prediction that are suitable for predicting partially conserved protein families. These two variants are shown to predict more accurately than the global prediction method. Our approach is also able to detect the boundaries of conserved domain/motifs in the entire sequences. The local prediction based on PST can be potentially applied to other problems as well. It is possible to modify these local prediction algorithms to extract unknown regulatory elements from DNA non-coding regions.

7 Acknowledgements

The authors wish to thank Drs. Bejerano and Dr. Yona for providing their PST program. We also thank Dr. Stephen Scott for his helpful discussion and Dr. Donald Weeks for his support.

References

Apostolico, A. & Bejerano, G. (2000): Optimal amnesic Probabilistic Automata or How to learn and classify proteins in linear time and space. *J. Comp. Biol*, 7(3/4):381-393.

Bailey, T. L. & Elkan, C. (1995): The value of prior knowledge in discovering motifs with MEME. *ISMB*, 3:21-29.

Bateman, A., Birney, E., Durbin, R., Eddy, E. R., Finn, R. D. & Sonnhammer, E. L. (1999): Pfam3.1: 1313 multiple alignments and profile HMMs match the majority of proteins *Nucleic Acids Res.*, 27:260-262.

Bates, P. A. and Sternberg, M. J. E. (1991): Modeling building by comparison at CASP3: using expert knowledge and computer automation. *Protein*, (Suppl.3):47-54.

Bejerano, G. and Yona, G. (2000): Variations on probabilistic suffix trees: statistical modelling and prediction of protein families. *Bioinformatics*, 17:23-43.

Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998): Biological sequence analysis. Chapter 2.

Gillman, D. & Sipser, M. (1994): Inference and minimization of hidden Markov chains. In *Proceedings of the 7th Annual Workship on Computational Learning Theory*, 147-158.

Gupta, M. & Liu, J. S. (2003): Discovery of conserved sequence patterns using a stochastic dictionary model. *Journal of the American Statistical Association*, 98(461):55-66.

Hofmann, K., Bucher, P., Falquet, L. & Bairoch, A. (1999): The PROSITE database, its status in 1999. *Nucleic Acids Res.*, 27:215-219.

Hughey, R. and Krogh, A. (1998): SAM: sequence alignment and modeling software system. *Technical Report UCSC-CRL-96-22* University of California, Santa Cruz, CA.

Im, C. S. and Grossman, A. R. (2001): Identification and regulation of high light-induced genes in *Chlamydonas reinhardtii*. *The Plant Journal*, 30(3):301-313.

Marchal, K., Thijs, G., De Keersmaecker, S., Monsieurs, P., De Moor, B., & Vanderleyden, J. (2003): Genome-specific higher-order background models to improve motif detection. *Trends Microbiol.*, 2003,11(2):61-6.

Pearson, W. R. (1995): Comparison of methods for searching protein sequence databases. *Protien Sci.*, 4:1145-1160.

Ron, D., Singer, Y. & Tishby, N. (1996): The power of amnesia: learning probabilistic automata with variable memory length. *Machine learning*, 25:117-149.

Thijs, G., Lescot, M., Marchal, K., Rombauts, S., De Moor, B., Rouze, P., & Moreau, Y. (2001): A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, 2001, 17:1113-1122.