CHAPTER 1

NEIGHBORHOOD PROFILE SEARCH FOR MOTIF REFINEMENT

1.1 INTRODUCTION

Recent developments in DNA sequencing have allowed biologists to obtain complete genomes for several species. However, knowledge of the sequence does not imply the understanding of how genes interact and regulate one another within the genome. Many transcription factor binding sites are highly conserved throughout the sequences and the discovery of the location of such binding sites plays an important role in understanding gene interaction and gene regulation. Biological experiments to find these regulatory sites such as DNA footprinting, gel shift analysis and others [1] are tedious and time consuming. To complement these traditional experimental techniques, scientists have started to develop computational methods to identify these regulatory sites.

In this chapter, we develop a new computational approach to find the regulatory sites in an efficient manner. Although there are several variations of the motif finding algorithms, the problem studied in this chapter is defined as follows: without any previous knowledge of the consensus pattern, discover all the occurrences of the motifs and then recover a pattern for which all of these instances are within a given number of mutations (or substitutions)

[2]. Usually, every instance of a motif will have the same length but have slightly different sequence compositions. In this chapter, we consider a precise version of the motif discovery problem in computational biology as discussed in [3,4]. The planted (l,d) motif problem [4] considered here is described as follows: Suppose there is a fixed but unknown nucleotide sequence M (the *motif*) of length l. The problem is to determine M, given t sequences with t_i being the length of the i^{th} sequence and each one containing a planted variant of M. More precisely, each such planted variant is a substring that is M with exactly d point substitutions (see Fig. 1.1.). More details about the complexity of the motif finding problem is given in [2]. A detailed assessment of different motif finding algorithms was published recently in [5].

Figure 1.1. Synthetic DNA sequences containing some instance of the pattern 'CCGATTACCGA' with a maximum number of 2 mutations. The motifs in each sequence are highlighted in the box. We have a (11,2) motif where 11 is the length of the motif and 2 is the number of mutations allowed.

Despite the significant amount of literature available on the motif finding problem, many do not exploit the probabilistic models used for motif refinement [6] [7]. More details on the estimates of the hardness of this problem without any complex information like overlapping motifs and background distribution is shown in [8]. We provide a novel optimization framework for refining motifs based on TRUST-TECH (TRansformation Under STability-reTaining Equilibria CHaracterization) methodology that can systematically explore subspace and perform neighborhood search effectively [9]. The rest of this chapter is organized as follows: Section 1.2 gives some relevant background about the existing approaches used for finding motifs. Section 1.3 describes the problem formulation and the expectation maximization algorithm in detail. Section 1.4 discusses our new framework and Section 1.5 details our implementation. Section 1.6 gives the experimental results from running our algorithm on synthetic and real datasets. Finally, Section 1.7 concludes our discussion with future research directions.

1.2 RELEVANT BACKGROUND

Existing approaches used to solve the motif finding problem can be classified into two main categories [10]. The first group of algorithms utilizes a generative probabilistic representation of the nucleotide positions to discover a consensus DNA pattern that maximizes the likelihood score. In this approach, the original problem of finding the best consensus pattern is formulated as finding the global maximum of a continuous non-convex function. The main advantage of this approach is that the generated profiles are highly representative of the signals being determined [11]. The disadvantage, however, is that the determination of the "best" motif cannot be guaranteed and is often a very difficult problem since finding global maximum of any continuous non-convex function is a challenging task. Current algorithms converge to the nearest local optimum instead of the global solution. Gibbs sampling [12], MEME [7], greedy CONSENSUS algorithm [13] and HMM based methods [14] belong to this category.

The second group uses patterns with 'mismatch representation' which defines a signal to be a consensus pattern and allows up to a certain number of mismatches to occur in each instance of the pattern. The goal of these algorithms is to recover the consensus pattern with the highest number of instances. These methods view the representation of the signals as discrete and the main advantage of these algorithms is that they can guarantee that the highest scoring pattern will be the global optimum for any scoring function. The disadvantage, however, is that consensus patterns are not as expressive of the DNA signal as profile representations. Recent approaches within this framework include Projection methods [3, 15], string based methods [4], Pattern-Branching [16], MULTIPROFILER [17], suffix trees [18] and other branch and bound approaches [10, 19]. Theoretically, the best approach for finding the consensus pattern is an exhaustive pattern driven search [20]. Since, the pattern search space grows exponentially, this approach is not feasible.

In summary, the consensus model is not a very good description of the functional sites. Some positions in a biologically functional site are much conserved than others and consensus model cannot represent these characteristics. The profile model, on the other hand, can reflect these characteristics in a better manner by treating each position differently with a different letter distribution. A hybrid approach could potentially combine the expressiveness of the profile representation with convergence guarantees of the consensus pattern. An example of a hybrid approach is the Random Projection [3] algorithm followed by the expectation maximization (EM) algorithm [7]. It uses a global solver to obtain promising

alignments in the discrete pattern space followed by further local solver refinements in continuous space [21, 22]. Currently, only few algorithms take advantage of a combined discrete and continuous space search [3, 10, 15]. In this chapter, we consider the profile representation of the motif and a new hybrid algorithm is developed to escape out of the local maximum of the likelihood surface obtained in this profile space. Some motivations to develop this new hybrid algorithm are :

- A motif refinement stage is vital and popularly used by many pattern based algorithms (like PROJECTION, MITRA etc) which try to find optimal motifs.
- The traditional EM algorithm used in the context of motif finding converges very quickly to the nearest local optimal solution (within 5-8 iterations) [23].
- There are many other promising local optimal solutions in the close vicinity of the profiles obtained from the global methods.

In spite of the importance placed on obtaining a global optimal solution in the context of motif finding, little work has been done in the direction of finding such solutions [24, 25]. There are several proposed methods to escape out of the local optimal solution to find better solutions in machine learning [26] and optimization [27] related problems. Most of them are stochastic in nature and usually rely on perturbing either the data or the hypothesis. These stochastic perturbation algorithms are inefficient because they will sometimes miss a neighborhood solution or obtain an already existing solution.

In this chapter, we develop TRUST-TECH based Expectation Maximization (TT-EM) algorithm and apply to the motif finding problem [9]. It has the capability to search for alignments corresponding to Tier-1 local maxima in the profile space in tier-by-tier manner systematically. This effective search is achieved by transforming the original optimization problem into a gradient system with certain properties and obtaining dynamical and topological properties of the gradient system corresponding to the nonlinear likelihood surface. Our method has been successfully used for obtaining the parameter estimates of finite mixture models [28]. The underlying theoretical details of our method are described in [29, 30].

1.3 PROFILE MODEL AND THE EM ALGORITHM

In this section, we will describe our problem formulation and the details of the EM algorithm in the context of motif finding problem. In the next section, we will describe some details

of the dynamical system of the log-likelihood function which enables us to search for the nearby local optimal solutions.

We will now transform the the problem of finding the best possible motif into a problem of finding the global maximum of a highly nonlinear log-likelihood scoring function obtained from its profile representation. The log-likelihood surface is made of $3l$ variables which are treated as the unknown parameters that are to be estimated. Here, we will describe these parameters ($Q_{k,j}$) and construct the scoring function in terms of these parameters.

Some promising initial alignments are obtained by applying projection methods or random starts on the entire dataset. Typically, random starts are used because they are cost efficient. The most promising sets of alignments are considered for further processing. These initial alignments are then converted into profile representation. Let t be the total number of sequences and $S = \{S_1, S_2...S_t\}$ be the set of t sequences. Let P be a single alignment containing the set of segments $\{P_1, P_2, ..., P_t\}$. l is the length of the consensus pattern. For further discussion, we use the following variables

$i = 1 ... t$ $\quad -$ for t sequences
$k = 1 ... l$ $\quad -$ for positions within an l-mer
$j \in \{A, T, G, C\}$ $\quad -$ for each nucleotide

j	$k=0$	$k=1$	$k=2$	$K=3$	$k=4$	...	$k=l$
A	$C_{0,1}$	$C_{1,1}$	$C_{2,1}$	$C_{3,1}$	$C_{4,1}$	...	$C_{l,1}$
T	$C_{0,2}$	$C_{1,2}$	$C_{2,2}$	$C_{3,2}$	$C_{4,2}$	...	$C_{l,2}$
G	$C_{0,3}$	$C_{1,3}$	$C_{2,3}$	$C_{3,3}$	$C_{4,3}$	...	$C_{l,3}$
C	$C_{0,4}$	$C_{1,4}$	$C_{2,4}$	$C_{3,4}$	$C_{4,4}$	...	$C_{l,4}$

Table 1.1. A count of nucleotides A, T, G, C at each position $K = 1..l$ in all the sequences of the data set. $K = 0$ denotes the background count.

The count matrix can be constructed from the given alignments as shown in Table 1.1. We define $C_{0,j}$ to be the overall background count of each nucleotide in all of the sequences. Similarly, $C_{k,j}$ is the count of each nucleotide in the k^{th} position (of the $l - mer$) in all the segments in P.

$$Q_{0,j} = \frac{C_{0,j}}{\sum_{J \in \{A,T,G,C\}} C_{0,J}} \tag{1.1}$$

$$Q_{k,j} = \frac{C_{k,j} + b_j}{t + \sum_{J \in \{A,T,G,C\}} b_J} \tag{1.2}$$

Eq. (1.1) shows the background frequency of each nucleotide, where b_j (and b_J) is known as the Laplacian or Bayesian correction and is equal to $d * Q_{0,j}$ and d is a constant usually set to unity. Eq. (1.2) gives the weight assigned to the type of nucleotide at the k^{th} position of the motif.

A Position Specific Scoring Matrix (PSSM) can be constructed from one set of instances in a given set of t sequences. In this model, every position of the motif is described as a probability distribution over the allowed alphabet. This model assumes that the alphabets are independently and identically distributed (i.i.d.) at each position and the starting positions at each site. It also assumes that these alphabets contribute additively to the total activity and hence the probability of an alignment matrix of n instances is determined by a multinomial distribution.

From (1.1) and (1.2), it is obvious that the following relationship holds:

$$\sum_{j \in \{A,T,G,C\}} Q_{k,j} = 1 \qquad \forall k = 0, 1, 2, ...l \tag{1.3}$$

For a given k value in (1.3), each Q can be represented in terms of the other three variables. Since the length of the motif is l, the final objective function (i.e. the likelihood score) would contain $3l$ independent variables. It should be noted that even if there are $4l$ variables in total, the parameter space will contain only $3l$ independent variables because of the constraints obtained from (1.3). Thus, the constraints help in reducing the dimensionality of the search problem.

To obtain the likelihood score, every possible $l - mer$ in each of the t sequences must be examined. This is done so by multiplying the respective $Q_{i,j}/Q_{0,j}$ dictated by the nucleotides and their respective positions within the $l - mer$. Only the highest scoring $l - mer$ in each sequence is noted and kept as part of the alignment. The total score is the sum of all the best (logarithmic) scores in each sequence.

$$A(Q) = \sum_{i=1}^{t} log(A)_i = \sum_{i=1}^{t} log \left(\prod_{k=1}^{l} \frac{Q_{k,j}}{Q_b} \right)_i = \sum_{i=1}^{t} \sum_{k=1}^{l} log(Q'_{k,j})_i \tag{1.4}$$

where $Q_{k,j}/Q_b$ represents the ratio of the nucleotide probability to the corresponding background probability. $Log(A)_i$ is the score at each individual i^{th} sequence. In equation (1.4), we see that A is composed of the product of the weights for each individual position k. We consider this to be the likelihood score which we would like to maximize. $A(Q)$ is the non-convex $3l$ dimensional continuous function for which the global maximum corresponds to the best possible motif in the dataset. More accurate motif models proposed in the literature [7] were not used here because they require significant additional calculations for fitting the parameters from the sequence data.

One of the primary tools for performing refinement of the candidate motifs by improving this likelihood score is the EM algorithm. We used the EM algorithm that has been proposed in [6]. Since, the position of the motif occurrence in each sequence is not fixed apriori, summing over all possible locations of motif instances becomes computationally tedious. EM algorithm overcomes this problem by iteratively seeking a better likelihood model and converges linearly to a locally maximum likelihood model from a given initial guess. Hence, this EM based refinement algorithm will seek a matrix model that locally maximizes the likelihood ratio. In other words, it tries to converge to a motif model that can explain the instances much better than the background model alone.

EM refinement performed at the end of a combinatorial approach has the disadvantage of converging to a local optimal solution. Many promising solutions might be in the close neighborhood in the model space. EM cannot obtain these models and thus outputs a sub-optimal solution. To avoid this problem, our method improves this procedure for refining the motifs by understanding the details of the stability boundaries of the likelihood function and deterministically tries to escape out of the convergence region of the EM algorithm.

1.4 NOVEL FRAMEWORK

In this section, we will describe the TRUST-TECH based expectation maximization algorithm that combines the advantages of both stability regions and the expectation maximization algorithm. Our framework consists of the following three stages:

- *Global stage* in which the promising solutions in the entire search space are obtained.

- *Refinement stage* (or *local stage*) where a local method is applied to the solutions obtained in the previous stage in order to refine the profiles.

- *Neighborhood-search stage* where the exit points are computed and the Tier-1 and Tier-2 solutions are explored systematically.

In the global stage, a branch and bound search is performed on the entire dataset. All of the profiles that do not meet a certain threshold (in terms of a given scoring function) are eliminated in this stage. Some promising initial alignments are obtained by applying these methods (like projection methods) on the entire dataset. Most promising set of alignments are considered for further processing. The promising patterns obtained are transformed into profiles and local improvements are made to these profiles in the refinement stage. The consensus pattern is obtained from each nucleotide that corresponds to the largest value in each column of the PSSM. The $3l$ variables chosen are the nucleotides that correspond to those that are not present in the consensus pattern. Because of the probability constraints discussed in the previous section, the largest weight can be represented in terms of the other three variables.

1.4.1 Hessian Matrix and Dynamical System for the Scoring Function

In order to present our algorithm, we have to define a dynamical system corresponding to the log-likelihood function and the PSSM. The key contribution of our work is the development of this nonlinear dynamical system which will enable us to realize the geometric and dynamic nature of the likelihood surface by allowing us to understand the topology and convergence behaviour of any given subspace on the surface. We construct the following *gradient system* in order to locate critical points of the objective function (1.4):

$$\dot{Q}(t) = -\nabla A(Q) \tag{1.5}$$

One can realize that this transformation preserves all of the critical points [29]. Now, we will present the details of the construction of the gradient system and the Hessian. In order to reduce the dominance of one variable over the other, the values of each of the nucleotides that belong to the consensus pattern at the position k will be represented in terms of the other three nucleotides in that particular column. Let P_{ik} denote the k^{th} position in the segment P_i. This will also minimize the dominance of the eigenvector directions when the Hessian is obtained. The variables in the scoring function are transformed into new variables described in Table 1.2. Thus, Eq. (1.4) can be rewritten in terms of the $3l$ variables as follows:

$$A(Q) = \sum_{i=1}^{t} \sum_{k=1}^{l} log\ f_{ik}(w_{3k-2}, w_{3k-1}, w_{3k})_i \tag{1.6}$$

where f_{ik} can take the values $\{w_{3k-2},\ w_{3k-1},\ w_{3k},\ 1-(w_{3k-2}+w_{3k-1}+w_{3k})\ \}$ depending on the P_{ik} value.

j	$k=b$	$k=1$	$k=2$	$K=3$	$k=4$	...	$k=l$
A	b_A	w_1	C_2	w_7	w_{10}	...	w_{3l-2}
T	b_T	w_2	w_4	w_8	C_4	...	w_{3l-1}
G	b_G	C_1	w_5	w_9	w_{11}	...	C_l
C	b_C	w_3	w_6	C_3	w_{12}	...	w_{3l}

Table 1.2. A count of nucleotides $j \in \{A,T,G,C\}$ at each position $k = 1 \ldots l$ in all the sequences of the data set. C_k is the k^{th} nucleotide of the consensus pattern which represents the nucleotide with the highest value in that column. Let the consensus pattern be GACT...G and b_j be the background.

The first derivative of the scoring function is a one dimensional vector with $3l$ elements.

$$\nabla A = \left[\frac{\partial A}{\partial w_1} \ \frac{\partial A}{\partial w_2} \ \frac{\partial A}{\partial w_3} \ \ldots . \ \frac{\partial A}{\partial w_{3l}} \right]^T \tag{1.7}$$

and each partial derivative is given by

$$\frac{\partial A}{\partial w_p} = \sum_{i=1}^{t} \frac{\frac{\partial f_{ip}}{\partial w_p}}{f_{ik}(w_{3k-2}, w_{3k-1}, w_{3k})} \tag{1.8}$$

$$\forall p = 1, 2 \ldots 3l \ \ and \ \ k = round(p/3) + 1$$

The Hessian $\nabla^2 A$ is a block diagonal matrix of block size 3X3. For a given sequence, the entries of the 3X3 block will be the same if that nucleotide belongs to the consensus pattern (C_k). This nonlinear transformation will preserve all the critical points on the likelihood surface. The theoretical details of the proposed method and their advantages are published in [29]. If we can identify all the saddle points on the stability boundary of a given local maximum, then we will be able to find all the tier-1 local maxima. Tier-1 local maximum is defined as the new local maximum that is connected to the original local maximum through one decomposition point. Similarly, we can define Tier-2 and Tier-k local maxima that will take 2 and k decomposition points respectively. However, finding all of the saddle points is computationally intractable and hence we have adopted a heuristic by generating the eigenvector directions of the PSSM at the local maximum. Also, for such a complicated likelihood function, it is not efficient to compute all saddle points on the

stability boundary. Hence, one can obtain new local maxima by obtaining the $exit\ points$ instead of the saddle points. The point along a particular direction where the function has the lowest value starting from the given local maximum is called the *exit point*.

1.4.2 TRUST-TECH based Framework

To solve Eq. (1.4), current algorithms begin at random initial alignment positions and attempt to converge to an alignment of $l-mer$s in all of the sequences that maximize the objective function. In other words, the $l-mer$ whose $log(A)_i$ is the highest (with a given PSSM) is noted in every sequence as part of the current alignment. During the maximization of $A(Q)$ function, the probability weight matrix and hence the corresponding alignments of $l-mer$s are updated. This occurs iteratively until the PSSM converges to the local optimal solution. The consensus pattern is obtained from the nucleotide with the largest weight in each position (column) of the PSSM. This converged PSSM and the set of alignments correspond to a local optimal solution. The neighborhood-search stage where the neighborhood of the original solution is explored in a systematic manner is shown below:

Input: Local Maximum (A).
Output: Best Local Maximum in the neighborhood region.
Algorithm:
Step 1: Construct the PSSM for the alignments corresponding to the local maximum (A) using Eqs. (1.1) and (1.2).
Step 2: Calculate the eigenvectors of the Hessian matrix for this PSSM.
Step 3: Find exit points (e_{1i}) on the practical stability boundary along each eigenvector direction.
Step 4: For each exit point, the corresponding Tier-1 local maxima (a_{1i}) are obtained by applying the EM algorithm after the ascent step.
Step 5: Repeat the above procedure for promising Tier-1 solutions to obtain Tier-2 neighborhood local maxima (a_{2j}).
Step 6: Return the solution that gives the maximum likelihood score amongst $\{A,\ a_{1i},\ a_{2j}\}$.

Fig. 1.2. illustrates the TRUST-TECH based EM (TT-EM) method. To escape out of this local optimal solution, our approach requires the computation of a Hessian matrix (i.e. the matrix of second derivatives) of dimension $(3l)^2$ and the $3l$ eigenvectors of the Hessian.

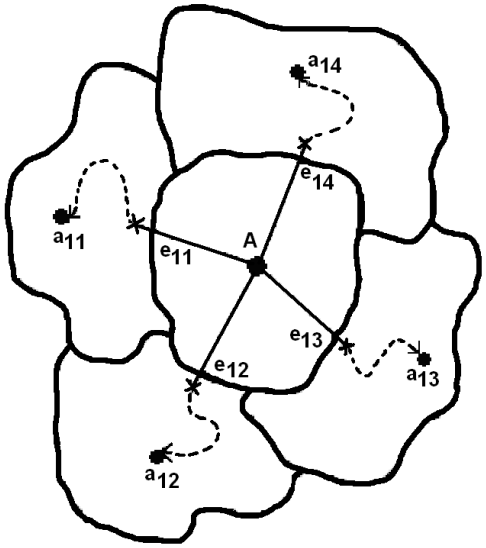

Figure 1.2. Diagram illustrates the TRUST-TECH method of escaping from the original solution (A) to the neighborhood local optimal solutions (a_{1i}) through the corresponding exit points (e_{1i}). The dotted lines indicate the local convergence of the EM algorithm.

these directions were chosen as a general heuristic and are not problem dependent. Depending on the dataset that is being worked on, one can obtain even more promising directions. The main reasons for choosing the eigenvectors of the Hessian as search directions are:

- Computing the eigenvectors of the Hessian is related to finding the directions with extreme values of the second derivatives, i.e., directions of extreme normal-to-isosurface change.
- The eigenvectors of the Hessian will form the basis vectors for the search directions. Any other search direction can be obtained by a linear combination of these directions.
- This will make our algorithm deterministic since the eigenvector directions are always unique.

The value of the objective function is evaluated along these eigenvector directions with some small step size increments. Since the starting position is a local optimal solution, one will see a steady decline in the function value during the initial steps; we call this the *descent stage*. Since the Hessian is obtained only once during the entire procedure, it is more efficient compared to Newton's method where an approximate Hessian is obtained for every iteration. After a certain number of evaluations, there may be an increase in the value

indicating that the stability boundary is reached. The point along this direction intersecting the stability boundary is called the *exit point*. Once the exit point has been reached, few more evaluations are made in the direction of the same eigenvector to improve the chances of reaching a new convergence region. This procedure is clearly shown in Fig 1.3.. Applying the local method directly from the exit point may give the original local maximum. The ascent stage is used to ensure that the new guess is in a different convergence zone. Hence, given the best local maximum obtained using any current local methods, this framework allows us to systematically escape out of the local maximum to explore surrounding local maxima. The complete algorithm is shown below :

Input: The DNA sequences, length of the motif(l), Number of Mutations(d)
Output: Motif (s)
Algorithm:
Step 1: Given the sequences, apply Random Projection algorithm to obtain different set of alignments.
Step 2: Choose the promising buckets and apply EM algorithm to refine these alignments.
Step 3: Apply the TT-EM method to obtain nearby promising local optimal solutions.
Step 4: Report the consensus pattern that corresponds to the best alignments and their corresponding PSSM.

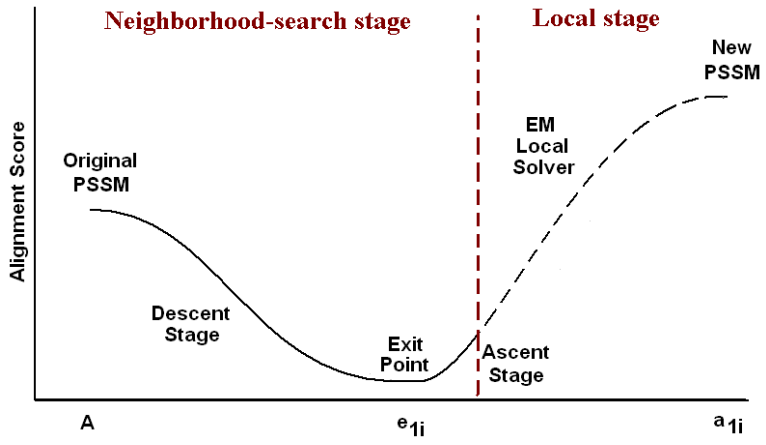

Figure 1.3. A summary of escaping out of the local optimum to the neighborhood local optimum. Observe the corresponding trend of $A(Q)$ at each step.

The new framework can be treated as a hybrid approach between global and local methods. It differs from traditional local methods by the ability to explore multiple local solutions in the neighborhood region in a systematic and effective manner. It differs from global methods by working completely in the profile space and searching a subspace efficiently in a deterministic manner. However, the main difference of this work compared to the algorithm presented in the previous chapter is that the global method is performed in discrete space and the local method is performed in the continuous space. In other words, the both the global and local method do not optimize the same function. In such cases, it is even more important to search for neighborhood local optimal solutions in the continuous space.

1.5 IMPLEMENTATION DETAILS

Our program was implemented on Red Hat Linux version 9 and runs on a Pentium IV 2.8 GHz machine. The core algorithm that we have implemented is TT_EM described in Algorithm 1. TT_EM obtains the initial alignments and the original data sequences along with the length of the motif. This procedure constructs the PSSM, performs EM refinement, and then computes the Tier-1 and Tier-2 solutions by calling the procedure $Next\_Tier$. The eigenvectors of the Hessian were computed using the source code obtained from [31]. $Next\_Tier$ takes a PSSM as an input and computes an array of PSSMs corresponding to the next tier local maxima using the TT-EM methodology.

Algorithm 1 Motif $TT\_EM(init\_aligns, seqs, l)$

$PSSM = Construct\_PSSM(init\_aligns)$
$New\_PSSM = Apply\_EM(PSSM, seqs)$
$TIER1 = Next\_Tier(seqs, New\_PSSM, l)$
for $i = 1$ to $3l$ **do**
  if $TIER1[i] <> zeros(4l)$ **then**
    $TIER2[i][\,] = Next\_Tier(seqs, TIER1[i], l)$
  end if
end for
Return $best(PSSM, TIER1, TIER2)$

Given a set of initial alignments, Algorithm 1 will find the best possible motif in the profile space in a tier-by-tier manner. For implementation considerations, we have shown only for two tiers. Initially, a PSSM is computed using $construct\_PSSM$ from the given

alignments. The procedure $Apply\_EM$ will return a new PSSM that corresponds to the alignments obtained after the EM algorithm has been applied to the initial PSSM. The details of the procedure $Next\_Tier$ are given in Algorithm 2. From a given local solution (or PSSM), $Next\_Tier$ will compute all the $3l$ new $PSSM$s corresponding to the tier-1 local optimal solutions. The second tier patterns are obtained by calling the $Next\_Tier$ from the first tier solutions. Sometimes, new PSSMs might not be obtained for certain search directions. In those cases, a zero vector of length $4l$ is returned. Only those new PSSMs which do not have this value will be used for any further processing. Finally, the pattern with the highest score amongst all the PSSMs is returned.

The procedure $Next\_Tier$ takes a PSSM, applies the TT-EM method and computes an array of PSSMs that corresponds to the next tier local optimal solutions. The procedure $eval$ evaluates the scoring function for the PSSM using (1.4). The procedures $Construct\_Hessian$ and $Compute\_EigVec$ compute the Hessian matrix and the eigenvectors respectively. MAX_iter indicates the maximum number of uphill evaluations that are required along each of the eigenvector directions. The neighborhood PSSMs will be stored in an array variable $PSSMs$ (this is a vector). The original PSSM is updated with a small step until an exit point is reached or the number of iterations exceeds the MAX_Iter value. Choosing an optimal step size is a heuristic. We choose the step size to be the average of the step values taken by the EM algorithm during its convergence. If the exit point is reached along a particular direction, few more function evaluations are made to ensure that the PSSM has exited the original convergence region and has entered a new one. The EM algorithm is then used during this ascent stage to obtain a new PSSM. For completeness, the entire algorithm has been shown in this section. However, during the implementation, several heuristics have been applied to reduce the running time of the algorithm. For example, if the first tier solution is not very promising, it will not be considered for obtaining the corresponding second tier solutions.

The initial alignments are converted into the profile space and a PSSM is constructed. The PSSM is updated (using the EM algorithm) until the alignments converge to a local optimal solution. The TT-EM methodology is then employed to escape out of this local optimal solution to compute nearby first tier local optimal solutions. This process is then repeated on promising first tier solutions to obtain second tier solutions. As shown in Fig. 1.2., from the original local optimal solution, various exit points and their corresponding new local optimal solutions are computed along each eigenvector direction. Sometimes, two directions may yield the same local optimal solution. This can be avoided by computing

Algorithm 2 PSSMs[ ] $Next\_Tier(seqs, PSSM, l)$

$Score = eval(PSSM)$
$Hess = Construct\_Hessian(PSSM)$
$Eig[\,] = Compute\_EigVec(Hess)$
$MAX\_Iter = 100$
for $k = 1$ to $3l$ **do**
  $PSSMs[k] = PSSM \quad Count = 0$
  $Old\_Score = Score \quad ep\_reached = FALSE$
  while (! $ep\_reached$) && ($Count < MAX\_Iter$) **do**
    $PSSMs[k] = update(PSSMs[k], Eig[k], step)$
    $Count = Count + 1$
    $New\_Score = eval(PSSMs[k])$
    if ($New\_Score > Old\_Score$) **then**
      $ep\_reached = TRUE$
    end if
    $Old\_Score = New\_Score$
  end while
  if $count < MAX\_Iter$ **then**
    $PSSMs[k] = update(PSSMs[k], Eig[k], ASC)$
    $PSSMs[k] = Apply\_EM(PSSMs[k], Seqs)$
  else
    $PSSMs[k] = zeros(4l)$
  end if
end for
Return $PSSMs[\,]$

the saddle point corresponding to the exit point on the stability boundary [32]. There can be many exit points, but there will only be a unique saddle point corresponding to the new local minimum. For computational efficiency, the TT-EM approach is only applied to promising initial alignments (i.e. random starts with higher likelihood score). Therefore, a threshold $A(Q)$ score is determined by the average of the three best first tier scores after 10-15 random starts; any current and first tier solution with scores greater than the threshold is considered for further analysis. Additional random starts are carried out in order to aggregate at least ten first tier solutions. The TT-EM method is repeated on all first tier solutions above a certain threshold to obtain second-tier local optimal solutions.

1.6 EXPERIMENTAL RESULTS

Experiments were performed on both synthetic data and real data. Two different methods were used in the global stage: random start and random projection. The main purpose of our work is not to demonstrate that our algorithm can outperform the existing motif finding algorithms. Rather, the main work here focuses on improving the results that are obtained from other efficient algorithms. We have chosen to demonstrate the performance of our algorithm on the results obtained from the random projection method which is a powerful global method that has outperformed other traditional motif finding approaches like MEME, Gibbs sampling, WINNOWER, SP-STAR, etc. [3]. Since the comparison was already published, we mainly focus on the performance improvements of our algorithm as compared to the random projection algorithm. For the random start experiment, a total of N random numbers between 1 and $(t - l + 1)$ corresponding to initial set of alignments are generated. We then proceeded to evaluate our TT-EM methodology from these alignments.

Fig. 1.4. shows the Tier-1 solutions obtained from a given consensus pattern. Since the exit points are being used instead of saddle points, our method might sometimes find the same local optimal solution obtained before. As seen from the figure, the Tier-1 solutions can differ from the original pattern by more than just one nucleotide position. Also, the function value at the exit points is much higher than the original value.

1.6.1 Synthetic Datasets

The synthetic datasets were generated by implanting some motif instances into $t = 20$ sequences each of length $t_i = 600$. Let m correspond to one full random projection + EM cycle. We have set $m = 1$ to demonstrate the efficiency of our approach. We compared

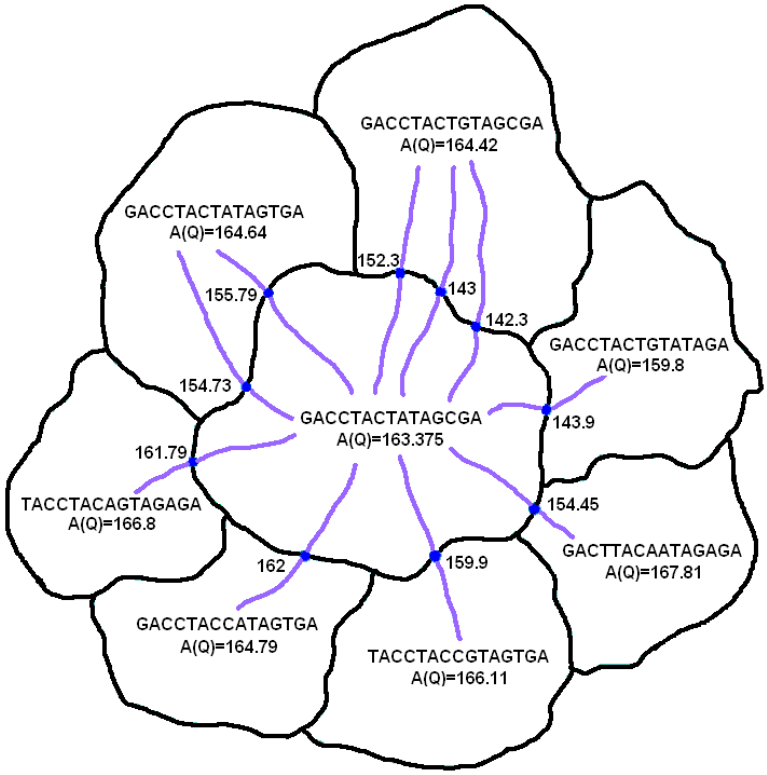

Figure 1.4. 2-D illustration of first tier improvements in a $3l$ dimensional objective function. The original local maximum has a score of 163.375. The various Tier-1 solutions are plotted and the one with highest score (167.81) is chosen.

the performance coefficient (PC) which gives a measure of the average performance of our implementation compared to that of Random Projection. The PC is given by :

$$PC = \frac{|K \cap P|}{|K \cup P|} \tag{1.9}$$

where K is the set of the residue positions of the planted motif instances, and P is the corresponding set of positions predicted by the algorithm. Table 1.4. gives an overview of the performance of our method compared to the random projection algorithm on the (*l*,*d*) motif problem for different *l* and *d* values.

Our results show that by branching out and discovering multiple local optimal solutions, higher m values are not needed. A higher m value corresponds to more computational time because projecting the l-mers into k-sized buckets is a time consuming task. Using our approach, we can replace the need for randomly projecting *l*-mers repeatedly in an effort to

converge to a global optimum by deterministically and systematically searching the solution space modeled by our dynamical system and improving the quality of the existing solutions. We can see that for higher length motifs, the improvements are more significant.

As opposed to stochastic processes like mutations in genetic algorithms, our approach eliminates the stochastic nature and obtains the nearby local optimal solutions systematically. Fig. 1.5. shows the performance of the TRUST-TECH approach on synthetic data for different (*l*,*d*) motifs. The average scores of the ten best solutions obtained from random starts and their corresponding improvements in Tier-1 and Tier-2 are reported. One can see that the improvements become more prominent as the length of the motif is increased. Table 1.3. shows the best and worst of these top ten random starts along with the consensus pattern and the alignment scores. We can see that, for higher length motifs, the improvements are more significant.

(l,d)	Initial Pattern	Score	First Tier Pattern	Score	Second Tier Pattern	Score
(11,2)	AACGGTCGCAG	125.1	CCCGGTCGCTG	147.1	CCCGGGAGCTG	153.3
(11,2)	ATACCAGTTAC	145.7	ATACCAGTTTC	151.3	ATACCAGGGTC	153.6
(13,3)	CTACGGTCGTCTT	142.6	CCACGGTTGTCTC	157.8	CCTCGGGTTTGTC	158.7
(13,3)	GACGCTAGGGGGT	158.3	GAGGCTGGGCAGT	161.7	GACCTTGGGTATT	165.8
(15,4)	CCGAAAAGAGTCCGA	147.5	CCGCAATGACTGGGT	169.1	CCGAAAGGACTGCGT	176.2
(15,4)	TGGGTGATGCCTATG	164.6	TGGGTGATGCCTATG	166.7	TGAGAGATGCCTATG	170.4
(17,5)	TTGTAGCAAAGGCTAAA	143.3	CAGTAGCAAAGACTACC	173.3	CAGTAGCAAAGACTTCC	175.8
(17,5)	ATCGCGAAAGGTTGTGG	174.1	ATCGCGAAAGGATGTGG	176.7	ATTGCGAAAGAATGTGG	178.3
(20,6)	CTGGTGATTGAGATCATCAT	165.9	CAGATGGTTGAGATCACCTT	186.9	CATTTAGCTGAGTTCACCTT	194.9
(20,6)	GGTCACTTAGTGGCGCCATG	216.3	GGTCACTTAGTGGCGCCATG	218.8	CGTCACTTAGTCGCGCCATG	219.7

Table 1.3. The consensus patterns and their corresponding scores of the original local optimal solution obtained from multiple random starts on the synthetic data. The best first tier and second tier optimal patterns and their corresponding scores are also reported.

With a few modifications, more experiments were conducted using the Random Projection method. The Random Projection method will eliminate non-promising regions in the search space and gives a number of promising sets of initial patterns. EM refinement is applied to only the promising initial patterns. Due to the robustness of the results, the TT-EM method is employed only on the top five local optima. The TT-EM method is again repeated on the top scoring first tier solutions to arrive at the second tier solutions. Fig. 1.6. shows the average alignment scores of the best random projection alignments and their corresponding improvements in tier-1 and tier-2 are reported. In general, the improvement in the first tier solutions are more significant than the improvements in the second tier solutions.

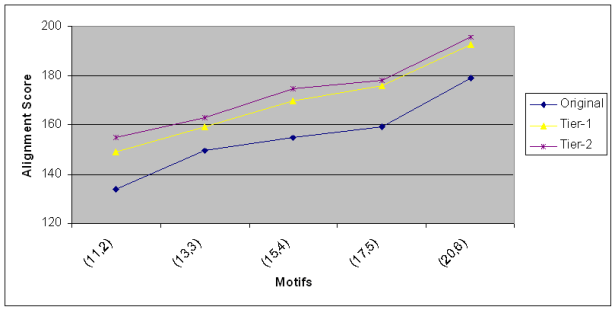

Figure 1.5. The average scores with the corresponding first tier and second tier improvements on synthetic data using the random starts with Exit-point approach with different (*l*,*d*) motifs.

Motif (l,d)	PC obtained using Random Projection	PC obtained using TT-EM method
(11,2)	20	20
(15,4)	14.875	17
(20,6)	12.667	18

Table 1.4. The results of performance coefficient with $m = 1$ on synthetically generated sequences. The likelihood scores are not normalized and the perfect score is 20 since there are 20 sequences.

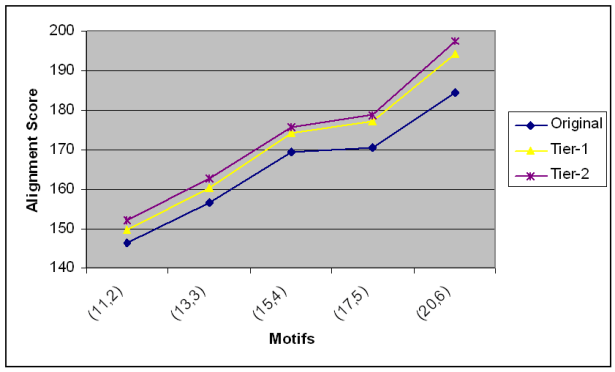

Figure 1.6. The average scores with the corresponding first tier and second tier improvements on synthetic data using the Random Projection with Exit-point approach with different (*l*,*d*) motifs.

Sequence	Sample Size	t	Best (20,2) Motif	Reference Motif
E. coli CRP	1890	18	TGTGAAATAGATCACATTTT	TGTGANNNNGNTCACA
preproinsulin	7689	4	GGAAATTGCAGCCTCAGCCC	CCTCAGCCC
DHFR	800	4	CTGCAATTTCGCGCCAAACT	ATTTCNNGCCA
metallothionein	6823	4	CCCTCTGCGCCCGGACCGGT	TGCRCYCGG
c-fos	3695	5	CCATATTAGGACATCTGCGT	CCATATTAGAGACTCT
yeast ECB	5000	5	GTATTTCCCGTTTAGGAAAA	TTTCCCNNTNAGGAAA

Table 1.5. Results of TT-EM method on biological samples. The real motifs were obtained in all the six cases using the TT-EM framework.

1.6.2 Real Datasets

Table 1.5. shows the results of the TT-EM methodology on real biological sequences. We have chosen $l = 20$ and $d = 2$. '*t*' indicates the number of sequences in the real data. For the biological samples taken from [3, 16], the value m once again is the average number of random projection + EM cycles required to discover the motif. All other parameter values (like projection size $k = 7$ and threshold s=4) are chosen to be the same as those used in the Random projection paper [3]. All of the motifs were recovered with $m = 1$ using the TT-EM strategy. Without our algorithm, the Random Projection algorithm needed *multiple cycles* (m=8 in some cases and m=15 in others) in order to retrieve the correct motif. This elucidates the fact that global methods can only be used to a certain extent and should be combined with refined local heuristics in order to obtain better efficiency. Since the random projection algorithm has outperformed other prominent motif finding algorithms like SP-STAR, WINNOWER, Gibbs sampling etc., we did not repeat the same experiments that were conducted in [3]. Running one cycle of random projection + EM is much more expensive computationally. The main advantage of our strategy comes from the deterministic nature of our algorithm in refining motifs.

Let the cost of applying EM algorithm for a given bucket be f and let the average number of buckets for a given projection be b. Then the running time of the TT-EM method will be $O(cbf)$ where c is a constant that is linear in l-the length of the motif. If there were m projections, then cost of the random projection algorithm using restarts will be $O(mbf)$. The two main advantages of using TT-EM strategy compared to random projection algorithm are :

- It avoids multiple random projections which often provide similar optimal motifs.
- It provides multiple optimal solutions in a promising region of a given bucket as opposed to a single solution provided by random projection algorithm.

1.7 CONCLUDING DISCUSSION

The TT-EM framework proposed in this chapter broadens the search region in order to obtain an improved solution which may potentially correspond to a better motif. In most of the profile based algorithms, EM is used to obtain the nearest local optimum from a given starting point. In our approach, we obtain promising results by computing multiple local optimal solutions in a tier-by-tier manner. We have shown on both real and synthetic data sets that beginning from the EM converged solution, the TT-EM approach is capable of searching in the neighborhood regions for another solution with an improved likelihood score. This will often translate into finding a pattern with less hamming distance from the resulting alignments in each sequence. Our approach has demonstrated an improvement in the score on all datasets that it was tested on. One of the primary advantages of the TT-EM methodology is that it can be used with different global and local methods. The main contribution of our work is to demonstrate the capability of this hybrid EM algorithm in the context of the motif finding problem. Our approach can potentially use any global method and improve its results efficiently. From our results, we observe that motif refinement stage plays a vital role and can yield accurate results deterministically. In the future, we would like to continue our work by combining other global methods available in the literature with existing local solvers like EM or GibbsDNA that work in continuous space. By following the example of [5], we may improve the chances of finding more promising patterns by combining our algorithm with different global and local methods.

REFERENCES

1. K. Docherty. *Gene Transcription, DNA Binding Proteins: Essential Techniques*. John Wiley and Sons, 1997.

2. P. Pevzner. *Computational Molecular Biology - an algorithmic approach*, chapter Finding Signals in DNA, pages 133–152. MIT Press, 2000.

3. J. Buhler and M. Tompa. Finding motifs using random projections. *Proceedings of the fifth annual international conference on Research in computational molecular biology*, pages 69–76, 2001.

4. P. Pevzner and S-H. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. *The Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 269–278, 2000.

5. M. Tompa and N. Li et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1):137 – 144, 2005.

6. C. E. Lawrence and A. A. Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function, and Genetics*, 7:41–51, 1990.

7. T. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *The First International Conference on Intelligent Systems for Molecular Biology*, pages 28–36, 1994.

8. M. Waterman, R. Arratia, and E. Galas. Pattern recognition in several sequences: consensus and alignment. *Mathematical Biology*, 46:515–527, 1984.

9. C. K. Reddy. *TRUST-TECH based Methods for Optimization and Learning*. PhD thesis, Cornell University, 2007.

10. E. Eskin. From profiles to patterns and back again: A branch and bound algorithm for finding near optimal motif profiles. *Proceedings of the eighth annual international conference on Research in computational molecular biology*, pages 115–124, 2004.

11. R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1999.

12. C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald, and J. Wootton. Detecting subtule sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

13. G. Hertz and G. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, 1999.

14. S. R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 1998.

15. B. Raphael, L.T. Liu, and G. Varghese. A uniform projection method for motif discovery in DNA sequences. *IEEE Transactions on Computational biology and Bioinformatics*, 1(2):91–94, 2004.

16. A. Price, S. Ramabhadran, and P.A. Pevzner. Finding subtle motifs by branching from sample strings. *Bioinformatics*, 1(1):1–7, 2003.

17. U. Keich and P. Pevzner. Finding motifs in the twilight zone. *Bioinformatics*, 18:1374–1381, 2002.

18. M. Sagot. Spelling approximate or repeated motifs using a suffix tree. *Lecture Notes in Computer Science*, 1380:111–127, 1998.

19. E. Eskin and P. Pevzner. Finding composite regulatory patterns in dna sequences. pages 354–363, 2002.

20. A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5(2):279–305, 1998.

21. Y. Barash, G. Bejerano, and N. Friedman. A simple hyper-geometric approach for discovering putative transcription factor binding sites. *Proc. of First International Workshop on Algorithms in Bioinformatics*, 2001.

22. E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From promoter sequence to expression: a probabilistic framework. In *Proceedings of the sixth annual international conference on Computational biology*, pages 263 – 272, Washington, DC, USA, 2002.

23. K. Blekas, D. Fotiadis, and A. Likas. Greedy mixture learning for multiple motif discovery in biological sequences. *Bioinformatics*, 19(5):607–617, 2003.

24. E. Xing, W. Wu, M.I. Jordan, and R. Karp. LOGOS: A modular bayesian model for de novo motif detection. *Journal of Bioinformatics and Computational Biology*, 2(1):127–154, 2004.

25. G. B. Fogel, D. G. Weekes, G. Varga, E. R. Dow, H. B. Harlow, J. E. Onyia, and C. Su1. Discovery of sequence motifs related to coexpression of genes using evolutionary computation. *Nucleic Acids Research*, 32(13):3826–3835, 2004.

26. G. Elidan, M. Ninio, N. Friedman, and D. Schuurmans. Data perturbation for escaping local maxima in learning. *In Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 132 – 139, 2002.

27. B. C. Cetin, J. Barhen, and J. W. Burdick. Terminal repeller unconstrained subenergy tunneling (TRUST) for fast global optimization. *Journal of Optimization Theory and Applications*, 77(1):97–126, 1993.

28. C. K. Reddy, H. D. Chiang, and B. Rajaratnam. Stability region based expectation maximization for model-based clustering. *in proceedings of sixth IEEE International Conference on Data Mining (ICDM)*, pages 522–531, 2006.

29. H. D. Chiang and C. C. Chu. A systematic search method for obtaining multiple local optimal solutions of nonlinear programming problems. *IEEE Transactions on Circuits and Systems: I Fundamental Theory and Applications*, 43(2):99–109, 1996.

30. J. Lee and H.D. Chiang. A dynamical trajectory-based methodology for systematically computing multiple optimal solutions of general nonlinear programming problems. *IEEE Transactions on Automatic Control*, 49(6):888 – 899, 2004.

31. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

32. C. K. Reddy and H. D. Chiang. A stability boundary based method for finding saddle points on potential energy surfaces. *Journal of Computational Biology*, 13(3):745–766, 2006.