# Improving Motif Refinement using Hybrid Expectation Maximization and Random Projection

Shashidhara H S*
Data Mining Laboratory
M S Ramaiah Institute of Technology, Bangalore
* Research Scholar, JNTU, Hyderabad
+91 80 23600822 151
shashi@msrit.edu

Prince Joseph
Data Mining Laboratory
M S Ramaiah Institute of Technology
Bangalore
+91 80 23600822 151
princejose1983@yahoo.com

K G Srinivasa
Data Mining Laboratory
M S Ramaiah Institute of Technology
Bangalore
+91 80 23600822 141
kgsrinivas@msrit.edu

## ABSTRACT
The main goal of the motif finding problem is to detect novel, over-represented unknown signals in a set of sequences. Popular algorithms like Expectation Maximization (EM) and Gibbs sampling are sensitive to the initial guesses and are known to converge to the nearest local maximum very quickly. A novel optimization framework searches the neighborhood regions of the initial alignments in a systematic manner to explore the multiple local optimal solutions. This effective search is achieved by transforming the original optimization problem into its corresponding dynamical system and estimating the practical stability boundary of the local maximum. The work aims at implementing the hybrid algorithm and enhancing it by trying different global methods and other techniques. Then aggregation methods rather than projection methods are tried.

## Categories and Subject Descriptors
J.3 [**Life and Medical Sciences**]: *Biology and genetics, Health and Medical Information Systems.*

## General Terms
Algorithms, Human Factors

## Keywords
Motif Finding, Expectation Maximization, Refinement, Random Projection

## 1. INTRODUCTION
Discovery of patterns in DNA sequences is one of the most challenging problems in molecular biology and computer science. The identification of regulatory motifs is essential for the study of gene expression. The main idea in gene expression is that every gene contains the information to produce a protein. Gene expression begins with binding of multiple protein factors, known as transcription factors, to enhancer and promoter sequences. Transcription factors regulate the gene expression by activating or inhibiting the transcription machinery.


*ISB'10*, February 15-17, 2010, Calicut, India.


The identification and discovery of regulatory elements using computational algorithms is difficult because they are frequently short and variable. The motif finding problem is the task of detecting overrepresented motifs as well as conserved motifs from the set of DNA sequences that are good candidates for being transcription factor binding sites. Transcription factor is a Protein that acts as regulator for gene expression, specifically regulating the activation of transcription process in which mRNA is made using DNA as a template. Motif is the common sequence "pattern" in the binding sites of a transcription factor. Finding motifs will help to develop disease treatments and understand disease susceptibility.

Synthesis of proteins is a two-step process. First step is *transcription* where an RNA "copy" of a portion of the DNA is synthesized. In the second step called *translation*, this RNA sequence is read and interpreted to synthesize a protein. Together, and these two steps are called *gene expression*. Gene expression and its regulation involve the binding of many regulatory transcription factors (TFs) to specific DNA elements called Transcription Factor Binding Sites (TFBS). In the last decade, the computational identification of TFBS (Transcription Factor Binding Sites) through the analysis of DNA sequence data has emerged as a major new technology to explain the transcription regulatory networks.

### 1.1 Motif Discovery Process
The mechanism that is responsible for the coordinated behavior of genes can be searched given a cluster of genes with highly similar expression profiles. It is assumed that co expression frequently arises from transcriptional co regulation. As co regulated genes are known to share some similarities in their regulatory mechanism, possibly at transcriptional level, their promoter regions might contain some common motifs that are binding sites for transcription regulators. A sensible approach to detect these regulatory elements is to search for statistically overrepresented motifs in the promoter region of such a set of co expressed genes.

In recent years, the efforts for large-scale sequencing of many genomes have lead to the easy availability of sequences that

contain regulatory elements. Technologies such as microarray and Chip-on-chip make it feasible to identify potential targets of transcription factors

In many organisms, the DNA that codes for proteins (genes) is only a small portion of the total genomic DNA. For example, genes make up only about 1.5% of the human genome. The control mechanism for activating and deactivating the genes is actually contained in the non coding components of DNA, which were initially considered as "junk" sequences. They take care of synthesis and non synthesis of proteins. Most of the control sequences for a gene lie in the *upstream regulatory region*, which is the region of a few thousand base pairs directly before the gene [also called the transcription regulatory region (TRR) or the promoter].

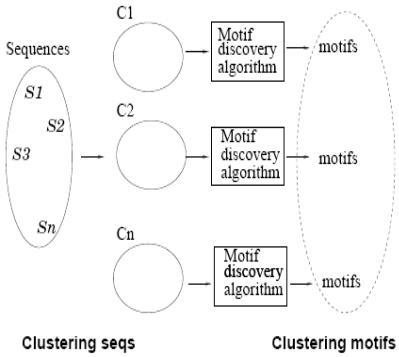


**Fig. 1. Showing the Motif discovery Process.**

## 1.2 Objectives of the Work

The objectives include experimenting with the hybrid motif discovery algorithms to improve the effectiveness, obtaining better parameters which would lead to good motifs in information content, generating the sequences which would closely resemble the real world sequences and finally implementing locality sensitive hashing method to achieve good multiple alignments.

# 2. RELATED WORK

## 2.1 Existing Methodologies

The present motif discovery algorithms are classified into:

(1) *Word-based (string-based) methods* that mostly rely on exhaustive enumeration, i.e., counting and comparing nucleotide frequencies and

(2) *Probabilistic sequence* models where the model parameters are estimated using maximum-likelihood principle or Bayesian inference.

The word-based enumerative methods guarantee global optimality and they are appropriate for short motifs. The word-based methods can also be very fast when implemented with optimized data structures such as suffix trees [1] and are a good choice for finding totally constrained motifs, i.e., all instances are identical. However, for typical transcription factor motifs that often have several weakly constrained positions, word-based methods can be problematic and the result often needs to be post-processed with some clustering system [2].

Probabilistic methods have the advantage of requiring few search parameters but rely on probabilistic models of the regulatory regions, which can be very sensitive with respect to small changes in the input data. Many of the algorithms developed from the probabilistic approach are designed to find longer or more general motifs than are required for transcription factor binding sites. However, these algorithms are not guaranteed to find globally optimal solutions.

Existing approaches used to solve the motif finding problem can be further classified into two main categories [3]. The first group of algorithms utilizes a generative probabilistic representation of the nucleotide positions to discover a consensus DNA pattern that maximizes information content score. In this approach, the original problem of finding the best consensus pattern is formulated into finding the global maximum of a continuous non-convex function.

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.

### *EM methods:*

EM for motif finding was introduced by Lawrence and Reilly [4] and it was an extension of the greedy algorithm for motif finding by Hertz *et al*. It was primarily developed for protein motifs; however, it can also be applied for DNA motif finding. No alignment of the sites is required and the basic model assumption is that each sequence must contain at least one common site. The uncertainty in the location of the sites is handled by employing the missing information principle to develop an EM algorithm. This approach allows for the simultaneous identification of the sites and characterization of the binding motifs. The MEME algorithm by Bailey and Elkan [5] extended the EM algorithm for identifying motifs in unaligned biopolymer sequences. The aim of MEME is to discover new motifs in a set of biopolymer sequences where little is known in advance about any motifs that may be present. MEME incorporated three novel ideas for discovering motifs. First, subsequences that actually occur in the biopolymer sequences are used as starting points for the EM algorithm to increase the probability of finding globally optimum motifs. Second, the assumption that each sequence contains exactly one occurrence of the shared motif is removed. Third, a method for probabilistically erasing shared motifs after they are found is incorporated so that several distinct motifs can be found in the same set of sequences; both when different motifs appear in different sequences and when a single sequence may contain multiple motifs.

### *Gibbs sampling methods:*

Among the probabilistic methods Gibbs sampling method has been used extensively for motif finding algorithms. Gibbs sampler for motif finding was developed by Lawrence *et al*. [6]. They did not apply this algorithm to DNA sequence but applied to protein sequence in the original article. Since one of the original assumptions of this algorithm was that there exists at

least one instance of a motif in every sequence, the method is sometimes called the "site sampler". Gibbs sampler is a Markov Chain Monte Carlo (MCMC) approach: "Markov-Chain", since the results from every step depends only on the results of the preceding one like in EM; "Monte-Carlo", since the way to select the next step is not deterministic but rather based on random sampling, i.e., random-numbers. The statistical background of MCMC methods is explained in the book by Liu [7] and that of Gibbs sampling in the article by Liu *et al*. [8]. In this algorithm it is assumed that we are given a set of $N$ sequences $S_1,...,S_N$ and we seek within each sequence mutually similar segments of specified width $W$.

## 2.2 Popular Motif Discovery Algorithms with Advantages and Limitations

*AlignACE* is a Gibbs sampling algorithm that returns a series of motifs as weight matrices that are over represented in the input set. AlignACE is the first statistical motif finder. It provides an adjunct measure that takes into account the sequence of the entire genome and highlights those motifs found preferentially in association with genes under consideration [9]

*GLAM* It is a Gibbs sampling based algorithm that automatically optimizes the alignment width and evaluates the stastical significance of its output [10]

*MEME* optimizes the E value of a statistic related to the information content of the motif. Rather than sum of information content of each motif column statistic used is the product of P values of column information contents [11] MEME use the EM algorithm. It is the most popular program for motif finding.

*BioProspector* is another variant of the Gibbs Sampling algorithm. Compared with the Lawrence version it added a Markov model estimated from all promoter sequences in the genome to model adjacent nucleotide dependency. It has 15 parameters. The default values for most of these parameters except for the motif width, which is set to 15, and the number of top motifs to report, which is set to 5. The background frequency model is generated using the whole genome, and the third-order Markov model is used unless otherwise specified. The order of the Markov model is chosen because it was the best among those tested [12].

# 3. PROPOSED WORK

## 3.1 Theoretical Background

X is said to be a *critical point* if it satisfies the following Condition

$$\nabla f(x) = 0 \quad \text{..... (4.1)}$$

The *saddle points* are critical points whose gradient is zero and Hessian of the nonlinear function has only one negative Eigen value. Intuitively, this means that a saddle point is a maximum along one direction but a minimum along all other orthogonal directions.

Negative *gradient system* is constructed in order to locate critical points of the objective function which is given by:

$$dx/dt = -\nabla f(x) \quad \text{..... (4.2)}$$

The solution curve of equation starting from x at time $t = 0$ is called a *trajectory* A state vector is called an *equilibrium point* of if $f(x) = 0$.

An equilibrium point is said to be hyperbolic if the Jacobian of f at point x has no eigenvalues with zero real part. A hyperbolic equilibrium point is called an asymptotically *stable equilibrium point (SEP)* if all the eigen values of its corresponding Jacobian have negative real part. Conversely, it is an unstable equilibrium point if some eigen values have a positive real part.

The stability region (also called region of attraction) of a stable equilibrium point xs of a dynamical system is denoted by *A(Xs)* and is given as

$$A(x_s) = \{x \in \Re^n : \lim_{t\to\infty} \Phi(x,t) = x_s\} \quad \text{..... (4.3)}$$

The **practical stability region** of a stable equilibrium point xs of a nonlinear dynamical system denoted by Ap(xs)

$$A_p(x_s) = \text{int}\,\overline{A(x_s)} \quad \text{……..…….... (4.4)}$$

A *type-1* equilibrium point $x_d$ *(k=1)* on the practical stability boundary of a stable equilibrium point $X_S$ is called a *decomposition point.*

## 3.2 Trust-Tech based Technique for Expectation Maximization [3]:

The EM algorithm is widely used for learning finite mixture models despite its greedy nature. Most popular model-based clustering techniques might yield poor clusters if the parameters are not initialized properly. To reduce the sensitivity of initial points, the proposed algorithm takes advantage of TRUST-TECH (Transformation under Stability Retaining Equilibrium Characterization) to compute neighborhood local maxima on the likelihood surface using stability regions. Basically, this method combines the advantages of the traditional EM with that of the dynamic and geometric characteristics of the stability regions of the corresponding nonlinear dynamical system of the log-likelihood function. More generic techniques like deterministic annealing [13], [14] and genetic algorithms [15], [16] have been applied to obtain a good set of parameters.

The global method gives some initial promising subspaces. The EM algorithm, along with the stability region phase, can obtain a set of promising neighborhood local maxima on the likelihood surface [1].

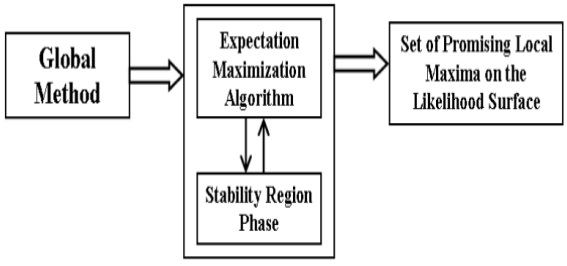


**Fig. 2. Block Diagram of TRUST-TECH Framework.**

The framework consists of three phases. The *first phase* is the global phase in which the promising solutions in the entire search space are obtained. The *second phase* is the local phase (or the EM phase), where the promising solutions obtained from the previous phase is refined to the corresponding locally optimal parameter set. The *third phase*, which is the main contribution of this paper, is the stability region phase. The exit points are computed and the neighborhood solutions are systematically explored through these exit points in this phase.

## 3.3 Problem Formulation

In it the problem of finding the best possible motif is transformed into a problem of finding the global maximum of a highly nonlinear log-likelihood scoring function obtained from its profile representation. Let $t$ be the total number of sequences and $n$ be the average length of each sequence. Let $S = \{S_1, S_2...S_t\}$ be the set of $t$ sequences. Let $P = \{P_1, P_2...P_t\}$ be the set of initial alignments. $l$ is the length of the consensus pattern

The count matrix can be constructed from the given alignments. We define $Q_{0,j}$ to be the non position specific background count of each nucleotide in all of the sequences where $j \in \{A, T,C,G\}$ is the running total of nucleotides occurring in each of the $l$ positions. Similarly, $Q_{k,j}$ is the count of each nucleotide in the $k^{th}$ position (of the $l-MER$) in all the $P$ alignments.

$$Q_{o,j} = C_{o,j} / \sum C_{o,j}$$
$$Q_{k,j} = C_{k,j} + b_j / N + \sum b_j \quad \text{.....(4.5)}$$

First equation shows the background frequency of each nucleotide where $b_j$ is known as the Laplacian or Bayesian correction and is equal to $d * Q_{0,j}$ where $d$ is some constant usually set to unity. Second equation gives the weight assigned to the type of nucleotide at the $k^{th}$ position of the motif. 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 information content score) would contain 3l independent variables.

A Position Specific Scoring Matrix (PSSM) can be constructed from one set of instances in a given set of $t$ sequences. Then compute the scoring function. To obtain the 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 scores in each sequence.

$$A(Q) = \sum_{i=1}^{t} \log(A)_i = \sum_{i=1}^{t} \log(\prod_{k=1}^{l} Q_{k,j} / Q_b)_i = \sum_{i=1}^{t} \sum_{k=1}^{l} \log(Q'_{k,j})_i \quad (4.6)$$

*In Eqn 4.6* $Q'_{k,j}$ is the ratio of the nucleotide probability to the corresponding background probability, i.e. $Q_{k,j}/Q_b$. $Log(A)_i$ is the score at each individual $i^{th}$ sequence where t is the total number of sequences. In above equation $A$ is composed of the product of the weights for each individual position $k$. $A(Q)$ is the non-convex $3l$ dimensional continuous function for which the global maximum corresponds to the best possible motif in the dataset. EM refinement that is done at the end of the combinatorial approaches has the main disadvantage that it converges to a local optimal solution. This method improves the refinement procedure by understanding the details about the stability boundaries and trying to escape out of the convergence region of the EM algorithm. In short problem of finding the optimal motif into a problem of finding the global maximum of a non-convex continuous $3l$ dimensional function

A gradient system is constructed order to locate critical points of the objective function. In order to reduce the dominance of one variable over the other, the values of the 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 $P_i$. The variables in the scoring function are transformed into new variables where $f_{ik}$ can take the values $\{\omega 3k-2, \omega 3k-1, \omega 3k, 1 - (\omega 3k-2 + \omega 3k-1 + \omega 3k)\}$ depending on the $P_{ik}$ value.

$$A(Q) = \sum_{i=1}^{t} \sum_{k=1}^{l} \log f_{ik}(\omega_{3k-2}, \omega_{3k-1}, \omega_{3k})_i \quad \text{..... (4.7)}$$

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

$$\nabla A = [\partial A / \partial \omega_1 \partial A / \partial \omega 2 ..... \partial A / \partial \omega_l]^T \quad \text{..... (4.8)}$$

And each partial derivative is given by:

$$\partial A / \partial \omega_p = \sum_{i=1}^{t} \partial f_{ip} / \partial \omega_p \; / \; f_{ik}(\omega_{3k-2}, \omega_{3k-1}, \omega_{3k}) \quad \text{.....(4.9)}$$

$$\forall_{p = 1, 2.....3l} \; And \quad k = round(p/3) + 1$$

The 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 The Hessian 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. If it is possible to identify all the saddle points on the stability boundary of a given local maximum to find all the tier-1 local maxima is easy However, finding all of the saddle points is computationally intractable and have adopted a heuristic by generating the eigenvector directions of the PSSM at the local maximum.

## 3.4 Algorithm

Input: The DNA sequences, length of the motif (l), Maximum Number of Mutations (d)

Output: Motif (s)

*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 exit point method to obtain nearby promising local optimal solutions.
*Step 4:* Report the consensus pattern that corresponds to the best alignments and their corresponding PSSM

The algorithm can pictorially be represented as:

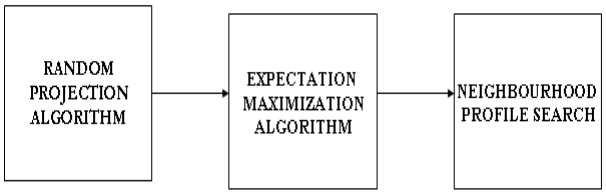


**Fig. 3. Showing the Block Diagram of Hybrid Method.**

## 3.5 Design Issues in Implementing the Random Projection Algorithm

Suppose a planted *(l, d)*-motif problem on t sequences of n letters. Randomly select k positions out of *l* positions. This selection is called *(l, k)* gapped pattern. For each *l-MER* in the data look at its *k-subset* which is defined by the gapped pattern. This subset is called projection.

Given a gapped pattern, count all *k*-length hashes of all *l-MERS*. Group *l-MERS* that hash to the same *k-MER* in a set called bucket. Random sequences should have the same bucket size for any hashed pattern. Some buckets will be significantly over-represented, which will mean that they represent motifs.

Like many probabilistic algorithms, the Projection algorithm performs a number of independent trials of a basic iteration. In each such trial, it chooses a random projection h and hashes each *l-MER x* in the input sequences to its bucket *h(x)*. Any hash bucket with sufficiently many entries is explored as a source of the planted motif, using a series of refinement steps

### *3.5.1 Finding the Planted Bucket*

The algorithm does not know which bucket is the planted bucket. So, it attempts to recover the motif from every bucket that contains at least s elements, where s is a threshold that is set so as to identify buckets that look as if they may be the planted bucket. In other words, the first part of the Projection algorithm is a heuristic for finding promising sets of *l-MERS* in the sequence. It must be followed by a refinement step that attempts to generate a motif from each such set.
Choose *k* of the *l* positions at random, without replacement. For an *l-MER x*, the hash function *h(x)* is obtained by concatenating the selected *k* residues of *x*. Viewing *x* as a point in *l*-dimensional Hamming space, *h(x)* is the projection of *x* onto a *k*-dimensional subspace. If *M* is the (unknown) motif, then bucket with hash value *h(M)* is the planted bucket.

The key idea is that, if $k < l - d$, then there is a good chance that some the *t* planted instances of *M* will be hashed to the planted bucket, namely all planted instances for which the *k* hash positions and *d* substituted positions are disjoint. So, there is a good chance that the planted bucket will be enriched for the planted motif, and will contain more entries then an average bucket

### *3.5.2 Choosing the Parameters*

The algorithm has three main parameters:

- The projection size *k*,
- The bucket (inspection) threshold *s*, and
- The number of independent trails *m*.

*Projection size:*
Ideally, the algorithm should hash a significant number of instances of the motif into the planted bucket, while avoiding contamination of the planted bucket by random background *l-MERS*.

To minimize the contamination of the planted bucket, choose k large enough. Since hashing $t\,(n - l + 1)$ *l-MERS* into $4k$ buckets, choose *k* such that $4k > t\,(n - l + 1)$, then the average bucket will contain less than one random *l-MER*.

*Bucket threshold:*
If the total amount of sequence is very large, then it may be that one cannot choose *k* to satisfy both $k < l - d$ and $4k > t\,(n - l + 1)$. In this case, set $k = l - d - 1$, as large as possible, and set the bucket threshold s to twice the average bucket size $t\,(n - l + 1)/4k$.

*Number of independent trails:*
Choose m so that the probability is at least $q = 0.95$ that the planted bucket contains s or more planted motif instances in at least one of the m trails.

## 3.6 Design Issues in Implementing EM Algorithm and Motif Refinement

The main loop of the Projection algorithm finds a set of buckets of *size* >= *s*. In the refinement step, each such bucket is explored in an attempt to recover the planted motif. The idea is that, if the current bucket is the planted buckets, then *k* of the planted motif residues are found. These, together with the remaining $l - k$ residues, should provide a strong signal that makes it easy to obtain the motif in only a few iterations of refinement.
Each bucket of *size* >= *s* is processed to obtain a candidate motif. Each of these candidates will be "refined" and the best refinement will be returned as the final solution. Candidate motifs are refined using the expectation maximization (EM) algorithm. This is based on the following probabilistic model:

An instance of some length-l motif occurs exactly once per input sequence. Instances are generated from a $4 \times l$ weight matrix model *W*, whose $(i, j)^{th}$ entry gives the probability that base *i* occurs in position *j* of an instance, independent of its other positions. The remaining $n-l$ residues in each sequence are chosen randomly and independently according to some background distribution.

Let S be a set of t input sequences, and let P be the background distribution. EM-based refinement seeks a weight matrix model $W^*$ that maximizes the likelihood ratio Pr(S | W*, P) and Pr(S | P), that is, a motif model that explains the input sequences much better than P alone.

The position at which the motif occurs in each sequence is not fixed a priori, making the computation of $W^*$ difficult, because Pr(S | W*, P) must be summed over all possible locations of the instances. To address this, the EM algorithm uses an iterative calculation that, given an initial guess $W_0$ at the motif model, converges linearly to a locally maximum-likelihood model in the neighborhood of W0. An initial guess $W_h$ for a bucket h is formed as follows: set $W_{h(i,\ j)}$ to the frequency of base $i$ among the $j^{th}$ positions of all *l-MERS* in *h*.

This guess forms a centroid for *h*, in the sense that positions that are well conserved in h are strongly biased in $W_h$, while poorly conserved positions are less biased. To avoid zero entries in $W_h$, add a Laplace correction of bi, to $W_{h(i,\ j)}$, where $b_i$ is the background probability of residue $i$ in the input. Once EM algorithm is used to obtain a refinement $W^*\ h$ of $W_h$, the final step is to identify the planted motif from $W^*\ h$. To do so, select from each input sequence the *l-MER x* with the largest likelihood ratio:

$$Pr(x \mid W^*\ h)\ Pr(x \mid P)$$

The resulting multiuse T of l-mer represents the motif in the input that is most consistent with $W^*\ h$. Let *CT* be the consensus of *T*, and let *s(T)* be the number of elements of *T* whose Hamming distance to *CT* is <= *d*. The algorithm returns the sequence *CT* that minimizes *s(T),* over all considered buckets *h* and over all trials.

## 3.7 Algorithm Projection with EM

Input: sequences $S_1, \ldots, S_t$, parameters k, s and m
Output: best guess motif
for $i = 1$ to $m$ do
  choose k different positions $Ik\ \{1, 2, \ldots, l\}$
for each *l-MER* $x \in S_1, \ldots, S_t$ do
  compute hash value $hIk\ (x)$
  Store $x$ in hash bucket
for each bucket with $>= S$ elements do
  refine bucket using EM algorithm
return consensus pattern of best refined bucket

## 3.8 Neighborhood Profile Search

The algorithm begin at random initial alignment positions and attempt to converge to an alignment of *l-MER* 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 – MERS* 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. It can be shown as:

1. Construct a PSSM (Position Specific Scoring matrix) from initial alignments.
2. Calculate eigenvectors of Hessian matrix.
3. Find exit points (or saddle points) along each eigenvector.
4. Apply EM from the new stability/convergence region. (Tier 1 Local Maxima)
5. Repeat first step. (Tier 2 Local Maxima)
6. Return max score {A, a1i, a2j}

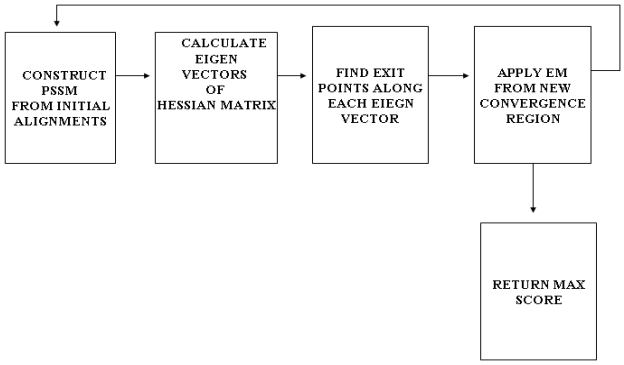


**Fig. 4. Highlighting the Proposed Method.**

# 4. RESULTS

The performance is evaluated using the performance coefficient, denoted as Let *K* denote the set of *t.l* base positions in the *t* occurrences of a planted motif, and let P denote the corresponding set of base positions in the *t* occurrences predicted by an algorithm. Then the algorithm's performance coefficient on the motif is denoted to be $K \cap P / K \cup P$. When all occurrences of the motif are found correctly, the performance coefficient achieves its maximum value of one. Table 4.1 below compares the performance of projection with that of previous motif discovery algorithms on sets of twenty random problem instances, each generated as described above.

**TABLE 1. Showing Performance Coefficients for Planted Motifs**

| *l* | *d* | GIBBS | WINNOWER | SP STAR | IMPROVED RANDOM |
|---|---|---|---|---|---|
| 10 | 2 | 0.20 | 0.78 | 0.56 | 0.82 |
| 11 | 2 | 0.68 | 0.90 | 0.84 | 0.91 |
| 12 | 3 | 0.03 | 0.75 | 0.33 | 0.81 |
| 13 | 3 | 0.60 | 0.92 | 0.92 | 0.92 |
| 14 | 4 | 0.02 | 0.02 | 0.20 | 0.77 |
| 15 | 4 | 0.19 | 0.92 | 0.73 | 0.93 |
| 16 | 5 | 0.02 | 0.03 | 0.04 | 0.70 |
| 17 | 5 | 0.28 | 0.03 | 0.69 | 0.93 |
| 18 | 6 | 0.03 | 0.03 | 0.03 | 0.74 |
| 19 | 6 | 0.05 | 0.03 | 0.40 | 0.96 |

Experimental results have shown that implementation is much more effective at recovering planted *(l; d)-motifs* in simulated data than existing algorithms. It has also proven effective in applications to real biological data.

## 5. CONCLUSION

The approach was an experiment to observe the proposed methodology's ability to improve the score which partially succeeded with the given samples. The problems observed were i) Difficult to distinguish spurious motifs from true ones, ii) Equal base frequencies were used in implementation; if an unequal frequency is tried performance drops, iii) When length of the sequence is increased, performance coefficient is reduced, iv) Refinement stage consumes more time affecting performance since most of the planted motifs will be hashed to buckets. Other than planted buckets also, performance of success rate is less for higher values of *l*. Basic improvements include predicting the length of the motif, finding multiple motifs in the same input automatically, and handling features such as spacers (sequences of N's) in the motif. A more challenging research problem is to extend the work to handle motifs whose instances contain insertions and deletions, which destroy the notion of fixed sequence positions used to define projections.

## REFERENCES


[1] Sagot M, 'Spelling Approximate Repeated or Common Motifs Using a Suffix Tree', *Lecture Notes in Computer Science,* 1380:111-127, 1998.

[2] Vilo J, Brazma A, Jonassen I, Robinson A, Ukonnen E, 'Mining for Putative Regulatory Elements in the Yeast Genome using Gene Expression Data', In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology AAAI Press San Diego, CA*, pp:384-394, 2000.

[3] 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*,115–124, 2004.

[4] Lawrence CE, Reilly A A, 'An Expectation Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences', *Proteins,* 1990, 7:41-51.

[5] Bailey TL, Elkan C, 'Unsupervised Learning of Multiple Motifs in Biopolymers using Expectation Maximization', *Machine Learning*, 21:51-80, 1995.

[6] Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC, 'Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment', *Science*, 262:208-214, 1993.

[7] Liu JS, 'Monte Carlo Strategies in Scientific Computing', *Springer Series in Statistics*; 2001.

[8] Liu J S, Neuwald A F, Lawrence CE, 'Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies', *J Amer Statist Assoc*, 90:1156-1170, 1995.

[9] Roth F.P., Hughes J.D., Estep P.W. and Church G.M, 'Finding DNA Regulatory Motifs within Unaligned Noncoding Sequences Clustered by Whole-Genome MRNA Quantization', *Nat. Biotechnol.*, 16, 939–945, 1998.

[10] Frith M C, Hansen U, Sponge J L, Weng Z, 'Finding Functional Sequence Elements by Multiple Local Alignment', *Nucleic acid Res* 32 189-200, 2004.

[11] Liu X, Brutlag D.L., Liu J S, 'BioProspector: Discovering Conserved DNA Motifs in Upstream Regulatory Regions of Co-expressed Genes', *Symposium on Biocomputin*g, 6, 127–138, 2001.

[12] Eskin E, Pevzner P, 'Finding Composite Regulatory Patternsin DNA Sequence', *Bioinformatics,* 18 : 354-363, 2002.

[13] K. Rose, 'Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems', *Proceedings. IEEE,* vol. 86, no. 11, pp. 2210-2239, 1998.

[14] N. Ueda and R. Nakano, 'Deterministic Annealing EM Algorithm', *Neural Networks, vol. 11,* no. 2, pp. 271-282, 1998.

[15] F. Pernkopf, D. Bouchaffra, 'Genetic-Based EM Algorithm for Learning Gaussian Mixture Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27*, no. 8, pp. 1344-1348, Aug. 2005.

[16] A.M. Martinez and J. Vitri, 'Learning Mixture Models Using a Genetic Version of the EM Algorithm', *Pattern Recognition Letters, vol. 21, no. 8*, pp. 759-769, 2000.