A PARTICLE SWARM OPTIMIZATION ALGORITHM FOR FINDING DNA SEQUENCE MOTIFS

Chengwei Lei
University of Texas at San Antonio, Department of Computer Science

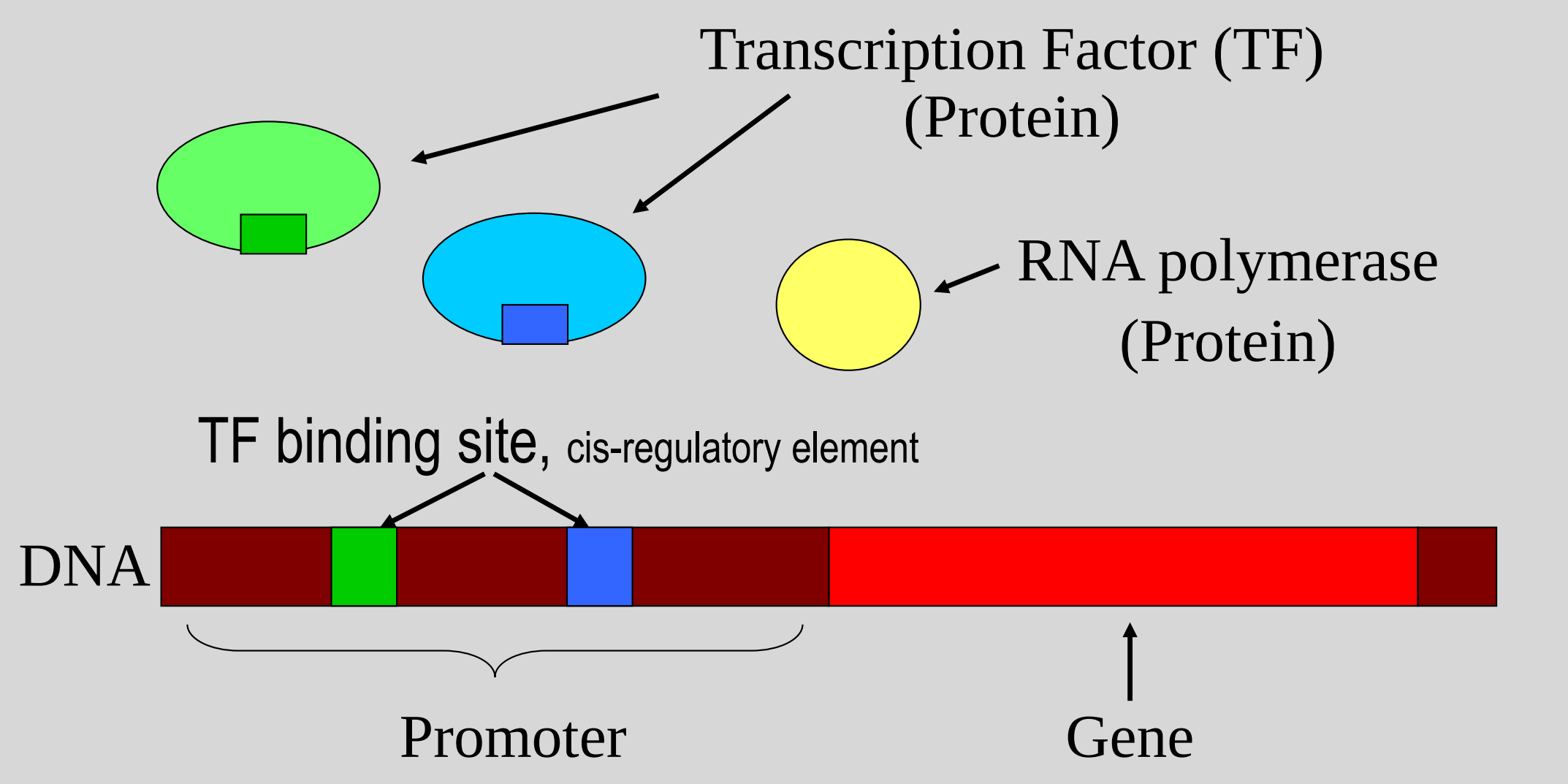
Introduction

Discovering short DNA motifs from a set of co-regulated genes is an important step towards deciphering the complex gene regulatory networks and understanding gene functions. Despite significant improvement in the last decade, it still remains one of the most challenging problems in both computer science and molecular biology.

Algorithm Background

Most of the computational approaches for finding motifs belong to one of two broad categories: stochastic optimization algorithms based on position specific weight matrices, or combinatorial search algorithms based on consensus patterns. Recent evaluation studies have shown that methods based on position specific weight matrices tend to stuck in local optima, while combinatorial search algorithms are typically limited to small data sets and short motifs only.

Biology Background


Transcription Factor (TF)
(Protein)
RNA polymerase
(Protein)
TF binding site, cis-regulatory element
DNA
Promoter
Gene

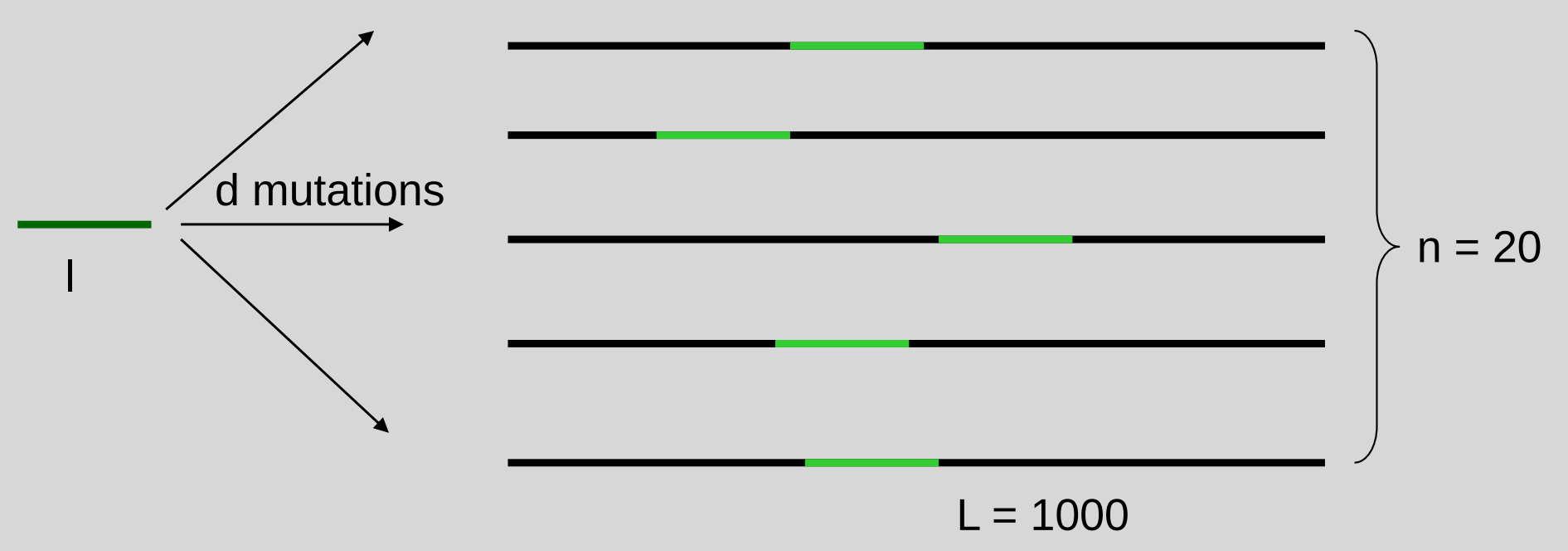
Challenging problem

The challenging problem is called *(l, d)-motif challenge problem* .

As shown in the figure, we try to find out the motif (green one) which hide in a bunch of sequences (black ones), and the motif in each sequences will have a few mutations(d bases).

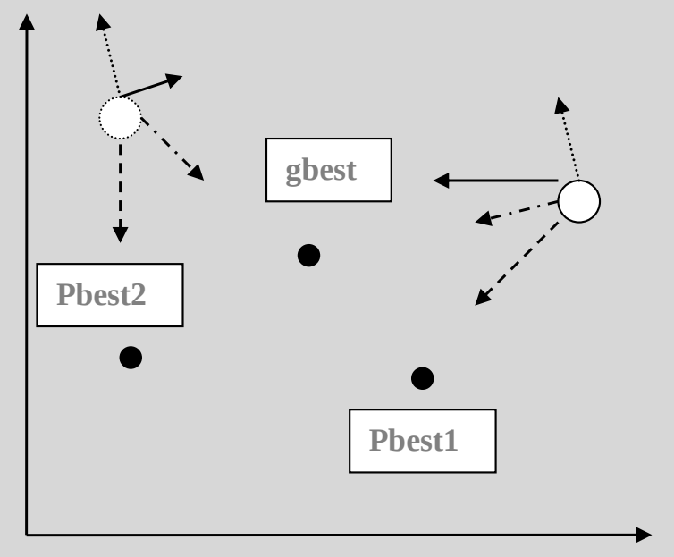
The previously algorithms cost too much memory or time to find out the result; my work is trying to find out a new algorithm use less memory and less time to find the motif.

Many algorithms fail at (15, 4)-motif for n = 20 and L = 1000



PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization (PSO) is a population based stochastic optimization technique and it is inspired by social behavior of bird flocking or fish schooling. It has been shown to be effective in optimizing difficult multidimensional problems in many fields.



$$V_n = \omega \cdot V_n + C_1 \cdot rand() \cdot (p_{best,n} - x_n) + C_2 \cdot rand() \cdot (g_{best,n} - x_n)$$

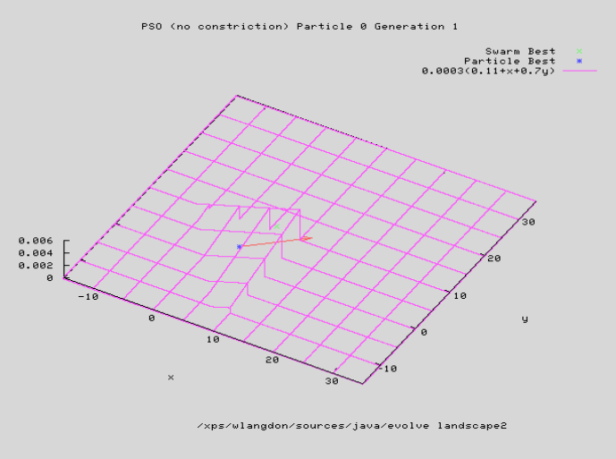
$$x_n = x_n + V_n$$

In PSO, each particle, is represented by a point in the multiple-dimensional solution space. Particles fly around for the optimal solution. During flight, each particle adjusts its position and velocity according to its own experience and the experiences of its neighbors. Each particle keeps track of the best solution(pbest) and the best solution by any particle in its neighborhood, which is the global optimum of all the particles, (gbest).

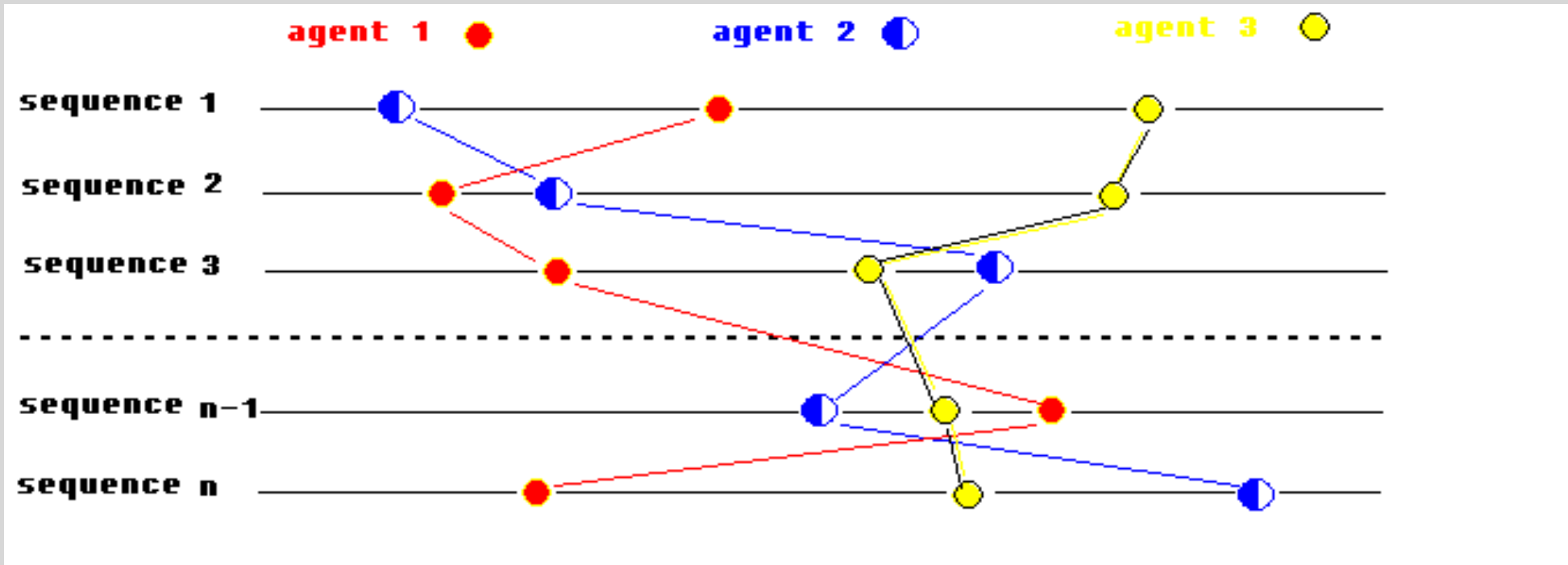
Difficulties for PSO in Motif Finding

The main difficulty associated with applying the PSO algorithm to motif finding problem is that the fitness function is not continuous.

In a typical PSO algorithm, one wishes to control the velocity so that at the beginning stage the particles can fly around quickly inside the search space, and when a particle approaches the optimal solution, it should slow down so it can converge. One can achieve this if the fitness function is continuous, since the velocity is updated according to the distances between the current position and the positions of pbest and gbest.



However, a solution in motif finding problem consists of a vector of positions in the input sequences.



Therefore, the distance between two potential solutions has no indication of the difference of their fitness values. A small change on the position will cause a totally different result.

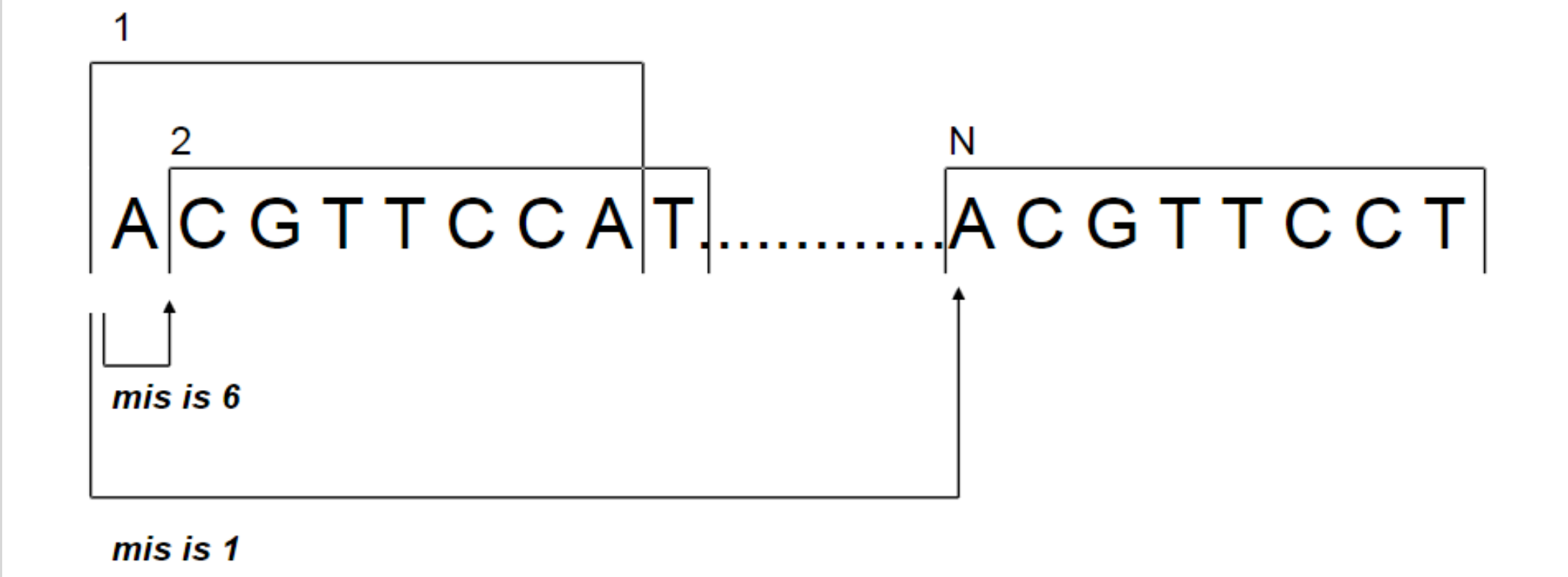
.........TACGATA.........
.........TAAAAT............
.........TATACT............
.........GATAAT............
.........TATGAT............
.........TATGTT............

PSO-motif algorithm

In order to apply PSO to the motif finding problem, we need to first define the solution structure and the fitness function. To evaluate the quality of a motif, we first derive a consensus for the motif by taking the most frequent base at each position, and then measure the total number of mismatches between the individual instances and the consensus.

Remap the neighborhood Information

The distance between two potential solutions has no indication of the difference of their fitness values. To solve this problem, we model the neighborhood information in the solution space by a dissimilarity graph in each sequence.



For example, given an input sequence CTCTGCTG and motif length = 3. We can built the mismatch table like this:

	P1	P2	P3	P4	P5	P6
P1	\	3	1	2	3	1
P2	3	\	3	2	1	3
P3	1	3	\	3	3	0
P4	2	2	3	\	3	3
P5	3	1	3	3	\	3
P6	1	3	0	3	3	\

Update policy

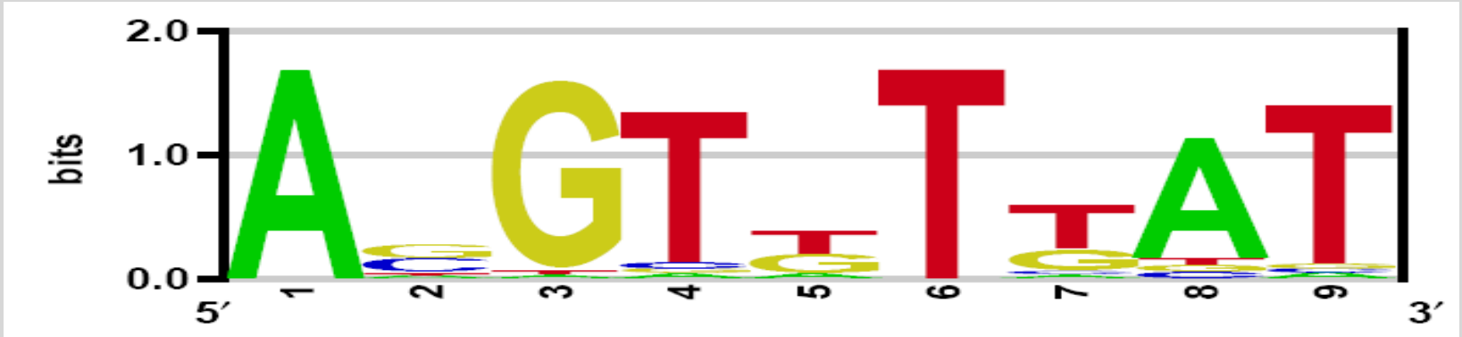
$$\mathbf{v}_i^u \leftarrow \omega \mathbf{v}_i^u + c_1^u \mathbf{r}_1 \circ \mathbf{D}(\mathbf{x}_i, \widehat{\mathbf{x}}_i) + c_2^u \mathbf{r}_2 \circ \mathbf{D}(\mathbf{x}_i, \mathbf{g})$$
$$\mathbf{v}_i^l \leftarrow \omega \mathbf{v}_i^l + c_1^l \mathbf{r}_1 \circ \mathbf{D}(\mathbf{x}_i, \widehat{\mathbf{x}}_i) + c_2^l \mathbf{r}_2 \circ \mathbf{D}(\mathbf{x}_i, \mathbf{g})$$

D (xi, xj) is the vector of numbers of mismatches between the motif instances in xi and xj

$$v_i^l(j) \leq D(x_i(i), x_i'(j)) \leq v_i^l(j).$$

Post-processing

By default, the quality of a motif is evaluated by the total number of mismatches between the consensus sequence and its instances. There are a number of limitations in this basic strategy when applied to real biological sequences.
We construct a position-specific weight matrix. The matrix is used to scan all input sequences. This will likely update some of the motif instances. We then re-compute the position specific weight matrix and repeat the scan, until the solution does not vary.



A	.97	.10	.02	.03	.10	.01	.05	.85	.03
C	.01	.40	.01	.04	.05	.01	.05	.05	.03
G	.01	.40	.95	.03	.40	.01	.3	.05	.03
T	.01	.10	.02	.90	.45	.97	.6	.05	.91

Results for Simulated Data Sets

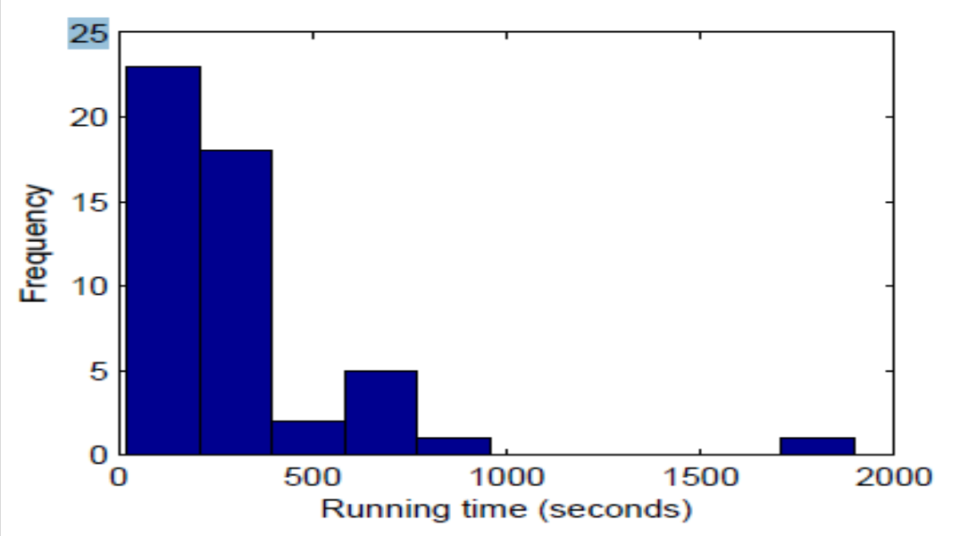
To objectively compare with the existing algorithms, we tested our algorithm on simulated data sets with the (l, d)-motif challenging problem.

Seq length	400	500	600	800	1000
Weeder	<1m	125s	200s	450s	15m
Projection	9s	23s	42s	162s	418s
PSO_mean	**18s**	**34s**	**57s**	**137s**	**288s**
PSO_mean	**7s**	**15s**	**36s**	**103s**	**220s**

Running time on (15,4)-problems. 20 sequences. Different seq length.

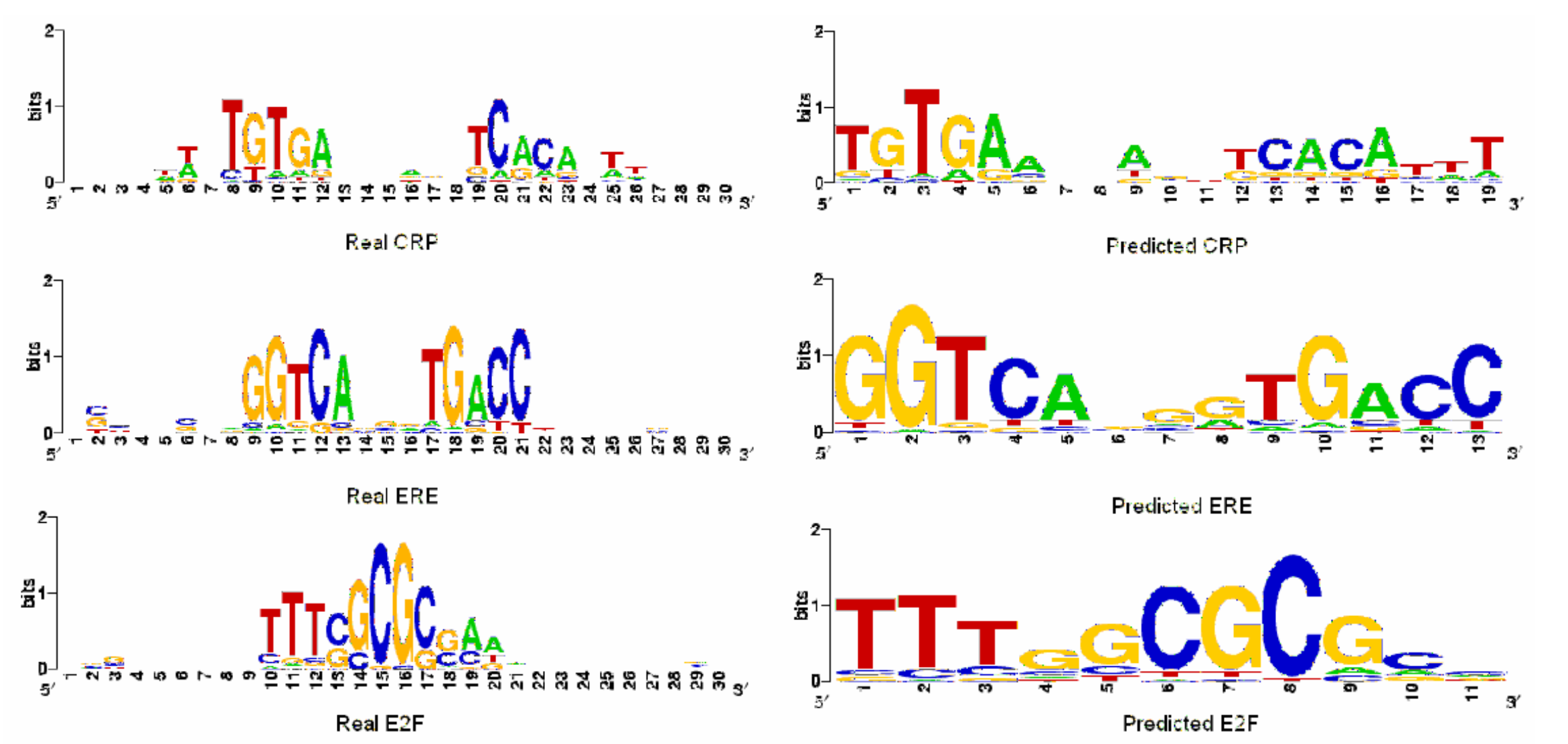
Running time on *(l,d)* problems. 20 sequences. Seq length is 600.

(*l*,*d*)	(11,2)	(13,3)	(15,4)	(17,5)	(19,6)
Projection	4s	13s	42s	94s	174s
MotifEnum	5s	119s	-	-	-
PSO_mean	**72s**	**58s**	**57s**	**61s**	**54s**
PSO_mean	**43s**	**48s**	**36s**	**38s**	**41s**



Distribution of running time
(15,4) problems
seq length = 1000.

Result for Real Motif



Conclusions

In this work, we have proposed a novel algorithm for finding DNA motifs based on a modified version of the Particle Swarm Optimization (PSO) algorithm. We have shown that we can successfully apply the PSO algorithm to solve the difficult motif finding problem with high efficiency and high accuracy.

Our experimental results on both simulated and real biological data sets are very encouraging. When applied to simulated challenge problems, PSO faster in difficult cases that have longer input sequences or longer motifs and more mutations; it does not require the number of mismatches to be given as a parameter, which is more useful in practice. For real biological sequences, our method combined with post-processing has successfully identified the known motifs and most of the binding sites.

Our studies have shown that PSO is a reliable and efficient technique for solving the difficult motif-finding problem, and we are looking into applying it to other challenging problems in computational biology.