NexusPSO: A Novel Algorithm to Detect Transcription Factor Binding Sites

Sarawoot Som-in, *Member, IAENG*, Warangkhana Kimpan

***Abstract*—The detection of transcription factor binding sites is a major problem in research in Biology. Methods and computer algorithms can be applied to reduce time complexity and cost of detecting transcription factor binding sites in laboratory experiments. One of the well-known methods commonly used is swarm intelligence. However, errors in detection of transcription factor binding sites can be caused by different binding sites in the same genome sequence. The purpose of this research is to improve the effectiveness and accuracy in the detection of transcription factor binding sites by applying the newly developed pre-processing procedure, Nexus, to Particle Swarm Optimization algorithm (NexusPSO). The accuracy of the NexusPSO algorithm was measured in comparison with other algorithms, using information content (IC) as an indicator, with *Escherichia coli* data. This study found that NexusPSO is the most accurate method being tested. NexusPSO was then tested using consensus sequences on *Saccharomyces cerevisiae* and *Homo sapiens*. NexusPSO showed nearly identical results when compared to DNA footprinting methods.**

***Index Terms*—Particle swarm optimization, Transcription factor binding site (TFBSs), Motif detection.**

I. Introduction

AMONG major DNA sequences component, there is conserved sequences fragment called Transcription factor binding sites (TFBSs). TFBSs are an integral part of the gene transcription process leading to protein synthesis. The TFBSs consist of subsequences known as motif sequences consisting of the same nucleotides: A, T, C and G. TFBSs assist the biological researchers in knowing the location of gene transcription which leads to protein synthesis. This information benefits researchers by reducing the cost, time and resources used in detecting TFBSs in the laboratory setting. TFBSs can be detected by employing rigorous labor using expensive laboratory equipment [1] resulting in high cost of experiments. Therefore, a computer application was developed to reduce the cost of detection by applying the Gibb Sampling algorithm, developed by Charles E. Lawrence et al [2]. Later, the Gibb Sampling algorithm was developed to detect TFBSs via online computing programs, including: AlignACE [3] and BioProspector [4]. Gibb Sampling algorithm consists of two main processes. The first process is the sampling step where random DNA sequences are sampled an analyzed for possible TFBSs. The data is input into a Position Weight Matrix (PWM). The PWM showed the probability of each alphabet (‘A’, ‘C’, ‘G’, ‘T’) in every position of the motif sequence. The second process is the predictive update step, where the full sequence of DNA is sampled, and the PWM is optimized and selects the most suitable motifs.

Gibb Sampling was further developed to detect TFBSs more effectively using software such as MEME [5], Weeder [6] and MDScan [7]. The Gibb sampling algorithm was then applied with the Bayesian probability model by Gibbs sampler [8]. Gibb Sampling is an algorithm classified as a type of searching or detecting method using statistical optimization. This was the most suitable technique of stochastic optimization suitable for searching in long sequences. However, the Gibb Sampling algorithm had limitation in terms of efficiency of time and accuracy.

The Genetic Algorithm (GA) was applied by Falcon F.M Liu et al. [9] to increase the efficiency of detecting motifs through a program called FMGA. This method can be applied to TFBSs. GA used a crossover technique to randomly process motif sequences for speed, and the mutation technique to generate quality PWM indicators in detection using SAGA [10], MDGA [11] algorithms.

When analyzing detection patterns of TFBSs, it can be considered a NP-Hard problem similar to the Traveling Salesman Problem (TSP) [12], Job-shop Scheduling Problem (JSP) [13], Flow Shop Scheduling Problem (FSP) [14], Longest Common Subsequence problem (LCS) [15], etc. [16]. Researchers have developed algorithms to solve NP-Hard problems such as Particle Swarm Optimization (PSO) algorithm [17] by J.Kennedy and R.Eberhart in 1995, the Ant Colony Optimization (ACO) algorithm by Dorigo et al. in 1996 [18], and Memetic algorithm by J. Yan and M. Li in 2015 [19]. However, such algorithms are still need to be improved as the problem of local optimums. These algorithms can be applied to detect TFBSs using hybrid concepts to avoid the problem of local optimums and/or to reduce time consumption of the algorithm process. Therefore, the algorithms were developed and applied for

Manuscript received April 12, 2018; revised June 9, 2018. This work was supported in part by King Mongkut’s Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand.

S. Som-in is with the Department of Computer Science, Faculty of Science, KMITL (Corresponding auther, phone: +662-329-8400, e-mail: fender.stad@gmail.com)

W. Kimpan is with the Department of Computer Science, Faculty of Science, KMITL (e-mail: warangkhana.ki@kmitl.ac.th)

exceeding these limitations such as time improvement in solving LCS problem using Simple Polynomial Time Algorithm [20] and improvement in both time and quality in detecting TFBSs using Ant Colony Regulatory Identification (ACRI) by Wei Liu et al. [21] and Particle Swarm Optimization Variants (PSO Variants) by Mustafa Karabulut and Turkay Ibrikci [22]. Both algorithms achieved admirable results, while ACRI can improve the speed of result, PSO can increase the accuracy. However, detection accuracy of TFBSs is still limited when detecting motif sequences containing different characteristics.

This paper proposes applying the PSO algorithm [17] and the newly developed Nexus procedure, called NexusPSO algorithm to yield more accurate results and avoid the problem of local optimums in detection of TFBSs. Nexus functions by creating custom subsequences in the genome sequence. Following the characterization of each subsequence, relationships are created within the subsequences. The quality of each relationship between subsequences is evaluated, and weak relationships pruned.

The remaining parts of this research are presented as follows: Section II discusses the problem domain and related work; Section III describes the proposed approach; the data set and experiments are explained in Section IV; and Section V is the conclusions.

II. Background and Related Theories

A. *Background and Signification of the Research Problem*

Detection of TFBSs can be considered a NP-Hard Problem. The variables of the problem can be defined as follows: The DNA sequences can be defined from the input sequence which is S_i where i is the sequence of any input sequence. While n is the total number of input sequences. The length of input sequence S_i is L_{S_i} and the length of motif sequences is w. The number of total motif sequences (number of M_{s_i}) in the input sequence S_i is number of M_{S_i} = L_{S_i}- $w + 1$ where $w < L_{S_i}$. The total number of input sequences are defined as $S = \{S_1, S_2,\dots,S_n\}$and the group of motif sequences in each input sequence is $S_i = \{M_1, M_2,\dots, M_{L-w-1}, M_{L-w}, M_{L+w+1}\}$. The group of total alphabet data possible in the genome sequences is $b = \{`A', `C', `G', `T'\}$.

If the detection of TFBSs independently allowed motif abundance, each sequence will be varied and the complexity would be $O((2^{l_i-w})^n$ [23],[24]. Therefore, restricting the number of motif in each particular sequence is preferred in this experiment.

B. *Definition of Particle Swarm Optimization (PSO)*

Particle Swarm Optimization (PSO) has been developed from the principles of swarm intelligence initiated from the research on the behaviors in movement in schools of birds or fish. While traveling, these groups vary group leaders to have the most effective leader at each iteration. Therefore, swarm intelligence has been developed by J.Kennedy and R.Eberhart in 1995 [17] as an algorithm for solving the NP-hard problems. This algorithm requires each bird or fish to be the considered a particle, with each particle selecting a different solution for each problem. Leaders are selected by running a fitness function and selecting the particle or particles with the highest calculated score.

One of the main principles of PSO is the definition of the particles. Then, topology is set, selecting the best particle at each iteration, including adjustments for speed and positions of each particle. The operation is repeated until each particle obtains the most optimal solution or the operation has reached the maximum iteration. There is also a research [25] which approaches the adjustment of particle speeds using Swap Sequence (SS) to achieve the better solutions.

The topology and connection among the particles within the PSO algorithm allow particles to share data according to the topography pattern. This causes each particle to move to a more suitable position by employing the data together among the best particles at each iteration within the neighborhood particles. The topologies [26] are as follows:

1. GBest: is the topology of total relative particles. Therefore, each particle has the number neighbors each particle has which is C_P *-1*, having C_P as the total number of particles as shown in Fig. 1(a).
2. Bidirectional Ring: is the topology of a ring with each particle having two neighboring particles: P_{i-1} and P_{i+1} when i is the current particle as shown in Fig. 1(b).
3. Random: is the random topology of non-structured relative particles as each particle chooses the neighbors by random and defines the number of neighbors C_n and $0 < C_n <= C_P -1$; particle as shown in Fig. 1(c).
4. Von Neumann: is the squared topology having the relative particles in a lattice structure. Each particle has four neighboring particles, consisting of: left P_{i-1} particle, right P_{i+1} particle, above $P_{s_{i-1}}$ particle, and below $P_{s_{i+1}}$ particle, as shown in Fig 1(d).

C. *Fitness Function for Accuracy Measurement*

The fitness function is run to consider and find the appropriate subsequences (appropriate motif sequences) that have the strongest solution. The factors used to calculate the fitness score of the particles or results of the motif consist of: equation (1) Consensus scoring (CS) [21] and equation (2) Information content (IC) [27]. CS is used to calculate the frequency of alphabetic patterns ‘A’, ‘C’, ‘G’ and ‘T’ in the results. This variable will not consider the frequency of other alphabets not involved in the motif sequences (background) as shown in Fig. 2. It is possible that high scores from CS can be attributed to background levels that are not accounted for in the score.

$$CS = 2 - (1/W)\sum_{i=1}^{W}\sum_{b=\{A,C,G,T\}} p_{bi} log_2(p_{bi}) \tag{1}$$

- b refers to all possible alphabets.
- w is the length of motif sequence.

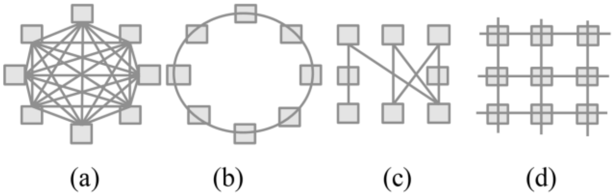


Fig. 1. Network pattern of Particle Swarm Optimization (PSO), Fig. (a) GBest. (b) Bidirectional Ring (c) Random and (d) Von Neumann.

- p_{bi} is the frequency of alphabet b.

Equation (2) Information Content (IC) is the variable used in calculating the similarity value of the alphabetic patterns between the results of each motif sequence. This variable will consider the frequency of other alphabets not involved in the results of motif sequences (background) as well.

$$IC = \sum f_b log_2 (f_b/p_b) \qquad (2)$$

- b refers to all possible alphabets.
- f_b is the frequency of alphabet b in any motif sequence.
- p_b is the frequency of alphabet b which is not in the results of motif sequences (background).

The best particle from all iterations is p_{best} and g_{best} is the best particle in the neighborhood from each iteration. p_{best} and g_{best} are the center in which the particle's neighborhood are required to move along, at different speeds depending on the distance of each particle relative to p_{best} and g_{best}. Considering equation (3), as the positions of particle p_i which is distance from particle p_{best} and particle g_{best} increase, particle speed will increase. On the contrary, p_i speed decreases the more near it draws to particles p_{best} and g_{best}.

$$v_{i+1}= w_i.v_i + c_1 y_i (x_{Pbest_i} - x_{Pi})+ c_2 z_i (x_{Gbest_i} - x_i) \qquad (3)$$

$$x_{i+1}= x_i + v_{i+1} \qquad (4)$$

The variables in the equations (3), (4) are as follows:

- w_i is the internal factor influencing the speed of particle p_i in the next generation v_{i+1}.
- c_1 ,c_2 is the value gained at random being from 0 to 1.
- x_{Pbest_i} is the best position from the previous functional round.
- x_{Gbest_i} is the best position from the group at each iteration with the definition as follows:

$$x_{Gbest_i} = arg\ min\ f\ (x*) = \{x* \in P : f\ (x*) <= f\ (x), \forall x \in I\}$$

- y_i and z_i is the parameter influencing the speed of particle p_i.

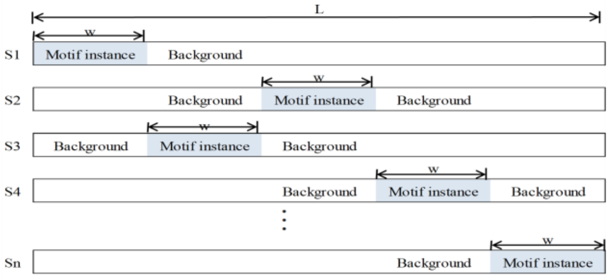


Fig. 2. Motif and background.

- v_i is the velocity at each iteration.
- x_{Pi} is the position of particle p_i at each iteration.

III. Proposed Principles and Concepts

The Nexus algorithm, which is a pre-process newly invented, can be applied to the PSO algorithm to increase the effectiveness in detecting TFBSs by reducing the chance of adhering to local optimums. The Nexus algorithm is able to reduce the problem space, which reduces the number of all possible subsequences, while still maintaining accurate results.

The Nexus algorithm consists of: grouping which will be stated descriptively in Section B; connection between the particles which will be stated descriptively in Section C; and the selection which will be stated descriptively in Section D.

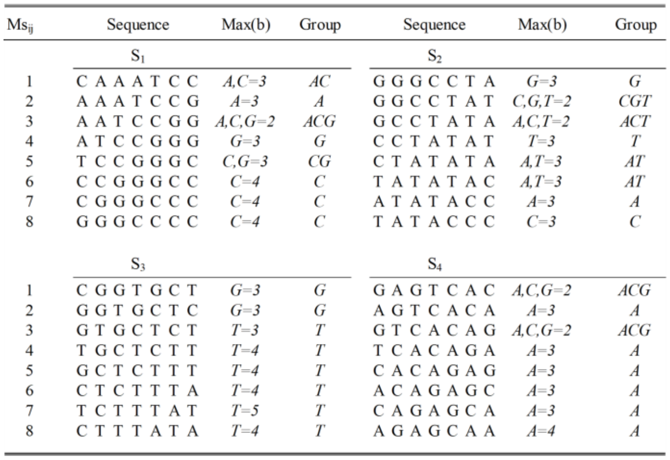
A. Indication of Variables

In this research, all input sequences are defined as $S = \{S_1, S_2, ..., S_n\}$ and n is the number of input sequences. Each sequence of S_i has the equal length L. Each subsequence in the input sequences S has equal length w. Any non-selected area is called background as shown in Fig. 2. The members of motif sequences, which are TFBSs $CoM = \{M_{S_1}, M_{S_2}, ..., M_{S_{n-1}}, M_{S_n}\}$. The members of alphabet or nucleotides b = {'A', 'C', 'G', 'T'}. All subsequences in each input sequence $S_i = \{M_1, M_2, ..., M_{L-w-1}, M_{L-w}, M_{L-w+1}\}$. Therefore, the total number of subsequences in the genome sequence is $(L-w+1)*S$

B. Grouping

This method uses a grouping procedure, which is the arrangement of subsequence into 4 groups following the number of members of N_s consisting of Group A, Group C, Group G, and Group T. Measuring counts the number of alphabets in each subsequence $M_{S_{ij}}$ from the total number of subsequences when any $M_{S_{ij}}$ has the maximum frequency of alphabet b $MAX(b)$ having $b \epsilon N_s$. Therefore $M_{S_{ij}}$ is classified into $Group(b)$. In the case that any $M_{S_{ij}}$ subsequence has the maximum frequency of alphabet $b > 1$, the subsequence $M_{S_{ij}}$ can be grouped into more than one group according to the maximum number of alphabet b as shown Table I. Table I shows grouping of subsequences with a total of 4 input

TABLE I
EXAMPLE OF INPUT SEQUENCE RESULTS FROM GROUPING



Ms_{ij}	Sequence	Max(b)	Group	Sequence	Max(b)	Group
	S_1			S_2		
1	C A A A T C C	*A,C=3*	*AC*	G G G C C T A	*G=3*	*G*
2	A A A T C C G	*A=3*	*A*	G G C C T A T	*C,G,T=2*	*CGT*
3	A A T C C G G	*A,C,G=2*	*ACG*	G C C T A T A	*A,C,T=2*	*ACT*
4	A T C C G G G	*G=3*	*G*	C C T A T A T	*T=3*	*T*
5	T C C G G G C	*C,G=3*	*CG*	C T A T A T A	*A,T=3*	*AT*
6	C C G G G C C	*C=4*	*C*	T A T A T A C	*A,T=3*	*AT*
7	C G G G C C C	*C=4*	*C*	A T A T A C C	*A=3*	*A*
8	G G G C C C C	*C=4*	*C*	T A T A C C C	*C=3*	*C*
	S_3			S_4		
1	C G G T G C T	*G=3*	*G*	G A G T C A C	*A,C,G=2*	*ACG*
2	G G T G C T C	*G=3*	*G*	A G T C A C A	*A=3*	*A*
3	G T G C T C T	*T=3*	*T*	G T C A C A G	*A,C,G=2*	*ACG*
4	T G C T C T T	*T=4*	*T*	T C A C A G A	*A=3*	*A*
5	G C T C T T T	*T=4*	*T*	C A C A G A G	*A=3*	*A*
6	C T C T T T A	*T=4*	*T*	A C A G A G C	*A=3*	*A*
7	T C T T T A T	*T=5*	*T*	C A G A G C A	*A=3*	*A*
8	C T T T A T A	*T=4*	*T*	A G A G C A A	*A=4*	*A*

sequences ($S_{n=4}$) having a total of possible $M_{S_{ij}}$ subsequences in the 8 input sequences. It can be noted the 1st sequence (S_1) is defined for more than one group because the maximum frequency of that subsequence matches more than one alphabet. Other examples include: the 1st subsequence ($M_{S_{11}}$), the 3rd subsequence ($M_{S_{13}}$), and the 5th subsequence ($M_{S_{15}}$), etc.

C. Connection

The creation of relations starts by taking all possible subsequences M_{ij} in the input sequence i to create relations with possible subsequences in the input sequence $i+1$ considering only subsequences in the same group (1< = i <= n-1 where n is the total number of input sequences). Therefore, the occurring pattern of relations between input sequence i and input sequence i+1 is as follows:

$$([M(A)_{ij} \bowtie M(A)_{i+1j}],[M(C)_{ij} \bowtie M(C)_{i+1j}],[M(G)_{ij} \bowtie M(G)_{i+1j}],[M(T)_{ij} \bowtie M(T)_{i+1j}])$$

$\bowtie$ is defined to be the related pairs of the subsequences in the input sequence S_i and S_{i+1}. The related pairs of the subsequences in the input sequence S_i and each input sequence S_{i+1} will define the CS value. The data in Table II shows an example of related pairs calculated as equation of CS as shown in equation (1).

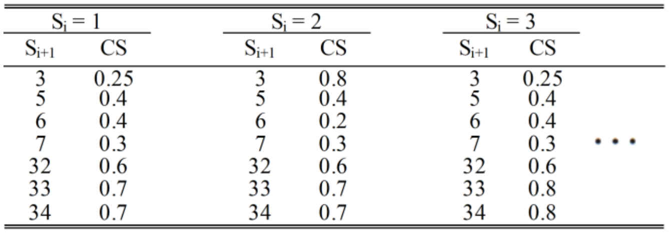
D. Selection

The selection of related pairs is the last process of the Nexus algorithm, where the best related pairs created in the connection process are selected. To select related pairs, the two input sequences with the highest CS value are selected.

$$[\text{Top2}\{M(A)_{ij} \bowtie M(A)_{i+1j}\},\text{Top2}\{M(C)_{ij} \bowtie M(C)_{i+1j}\},$$
$$\text{Top2}\{M(G)_{ij} \bowtie M(G)_{i+1j}\},\text{Top2}\{M(T)_{ij} \bowtie M(T)_{i+1j}\}]$$

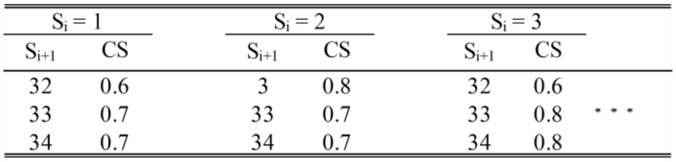
The example in Table III shows the related pairs being selected from the subsequence M_{ij} and subsequence M_{i+1j} being in the same group. The data in this table is selected from the data in Table II. This process is intended to reduce the problem of local optimums from the random PSO process.

TABLE II
EXAMPLE OF RELATED PAIRS BETWEEN SUBSEQUENCES IN THE INPUT SEQUENCE S_I AND S_{i+1}



S_i = 1		S_i = 2		S_i = 3		
S_{i+1}	CS	S_{i+1}	CS	S_{i+1}	CS	
3	0.25	3	0.8	3	0.25	
5	0.4	5	0.4	5	0.4	
6	0.4	6	0.2	6	0.4	
7	0.3	7	0.3	7	0.3	• • •
32	0.6	32	0.6	32	0.6	
33	0.7	33	0.7	33	0.8	
34	0.7	34	0.7	34	0.8	

TABLE III
EXAMPLE OF RELATED PAIRS SELECTED FROM THE INPUT SEQUENCE S_I AND S_{i+1}



S_i = 1		S_i = 2		S_i = 3		
S_{i+1}	CS	S_{i+1}	CS	S_{i+1}	CS	
32	0.6	3	0.8	32	0.6	
33	0.7	33	0.7	33	0.8	• • •
34	0.7	34	0.7	34	0.8	

E. Particles Initialization

The process of defining particles in the NexusPSO is through the creation of particle P_i in the swarm. Each particle P_i consists of subsequence M_{ij} (defining i and j as any input sequence and subsequence, respectively) from each input sequence S_i. The condition allows one subsequence per input sequence. This research defines the first input sequence S_1 to be the data sequence defining the first motif of each particle having S_1 ={ $M_{11}, M_{12},\ldots,M_{1L-w-1}, M_{1L-w}, M_{1L-w+1}$} where particle $P_{i(M_1)} = M_{1j}$. $P_{i(M_1)}$ is the first subsequence of the particle (initial subsequence) defining each particle P_i to select the subsequence from the next input sequence until the last data sequence is determined. Subsequences with the highest CS score are selected from the related pairs resulting in P_i= ($P_{i(M_1)}$, $P_{i(M_2)}$.., $P_{i(M_{n-1})}$, $P_{i(M_n)}$), where n is the total number of input data. The patterns of particle P_i in each group have created the related pairs as follows:

$P(A)_i$ any particle in group 'A'

$$[\text{Top1}\{M(A)_{1j} \bowtie (M(A)_{2j_{Top1}},M(A)_{2j_{Top2}})\}$$
$$\bowtie \text{Top1}\{M(A)_{2j_{Op}} \bowtie (M(A)_{3j_{Top1}},M(A)_{3j_{Top2}})\}$$
$$\vdots$$
$$\bowtie \text{Top1}\{M(A)_{n-1j_{Op}} \bowtie (M(A)_{nj_{Top1}},M(A)_{nj_{Top2}})\}]$$

$P(C)_i$ any particle in group 'C'

$$[\text{Top1}\{M(C)_{1j} \bowtie (M(C)_{2j_{Top1}},M(C)_{2j_{Top2}})\}$$
$$\bowtie \text{Top1}\{M(C)_{2j_{Op}} \bowtie (M(C)_{3j_{Top1}},M(C)_{3j_{Top2}})\}$$
$$\vdots$$
$$\bowtie \text{Top1}\{M(C)_{n-1j_{Op}} \bowtie (M(C)_{nj_{Top1}},M(C)_{nj_{Top2}})\}]$$

$P(G)_i$ any particle in group 'G'
$[\text{Top1}\{M(G)_{1j} \bowtie (M(G)_{2j_{Top1}}, M(G)_{2j_{Top2}})\}$
$\bowtie \text{Top1}\{M(G)_{2j_{Op}} \bowtie (M(G)_{3j_{Top1}}, M(G)_{3j_{Top2}})\}$
$\vdots$
$\bowtie \text{Top1}\{M(G)_{n\text{-}1j_{Op}} \bowtie (M(G)_{nj_{Top1}}, M(G)_{nj_{Top2}})\}]$

$P(T)_i$ any particle in group 'T'
$[\text{Top1}\{M(T)_{1j} \bowtie (M(T)_{2j_{Top1}}, M(T)_{2j_{Top2}})\}$
$\bowtie \text{Top1}\{M(T)_{2j_{Op}} \bowtie (M(T)_{3j_{Top1}}, M(T)_{3j_{Top2}})\}$
$\vdots$
$\bowtie \text{Top1}\{M(T)_{n\text{-}1j_{Op}} \bowtie (M(T)_{nj_{Top1}}, M(T)_{nj_{Top2}})\}]$

The meanings of symbols and variables are as follows:

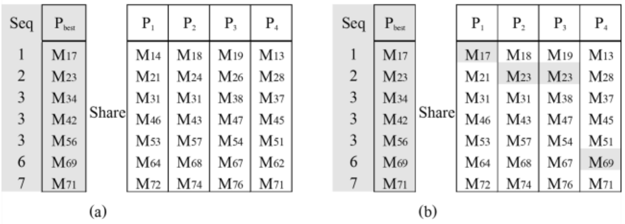
- $\bowtie$ is the relation of pairs in the sequences between the input sequence S_i with the input sequence S_{i+1}.
- $M(b)_{ij}$ is a subsequence in the genome defining i and j to be any input sequence and any subsequence, respectively. The set of b is {'A','C','G','T'}.
- $M(b)_{ij_{Top1}}$ is a subsequence with the highest CS value in relation to subsequence $M(b)_{i\text{-}1j}$.
- $M(b)_{ij_{Top2}}$ is the subsequence with the second highest CS value in relation to subsequence $M(b)_{i\text{-}1j}$.
- n is the total number of input sequences.
- $M(b)_{ij_{op}}$ is the optimal result of subsequences.

The particles of NexusPSO algorithm are defined to have the number of particles equal to the total possible subsequences of each input sequence L_i-w+1 with a size of $w < L_i$.

F. Particle's Movement

The initial position of the particles is defined in the process of initializing particles, as described in Section E. The NexusPSO defines the initial velocity of all particles as 0 and uses the fitness value from equation (5), which is discussed in Section G. This is used to calculate the fitness value of each particle. The fitness values from every particle are then compared to indicate the most suitable particle P_{best} as shown in Fig. 3(a). The comparison will be conducted by Gbest topology, with the topology using data shared among all particles, as shown in Fig. 1(a).

After, the position of each particle within the neighborhood is adjusted by applying the data of subsequence M_{ij} from the best particle P_{best} to replace the subsequences of particle's neighborhood P_i, as shown in Fig. 3(b). Adjusting the position of particles in each iteration, results in the particles having continuous movement, until each particle obtains the most optimal solution or the operation has reached the maximum iteration. If the process ceases because the total number of iterations was reached, the algorithm will select the particle with the highest fitness score from the last iteration. The results of the NexusPSO algorithm indicate the position of TFBSs in the genome sequences.



Seq	P_{best}		P_1	P_2	P_3	P_4
1	M17		M14	M18	M19	M13
2	M23		M21	M24	M26	M28
3	M34		M31	M31	M38	M37
3	M42	Share	M46	M43	M47	M45
3	M56		M53	M57	M54	M51
6	M69		M64	M68	M67	M62
7	M71		M72	M74	M76	M71

(a)

Seq	P_{best}		P_1	P_2	P_3	P_4
1	M17		M17	M18	M19	M13
2	M23		M21	M23	M23	M28
3	M34		M31	M31	M38	M37
3	M42	Share	M46	M43	M47	M45
3	M56		M53	M57	M54	M51
6	M69		M64	M68	M67	M69
7	M71		M72	M74	M76	M71

(b)

Fig. 3. Example of adjusting the particle position. (a) represents the 5 particles in the input sequences. (b) shows the replacements within the subsequences.

G. Fitness Function

The scale measuring the particles optimal P_{best} at each iteration t_i is the fitness function. The NexusPSO algorithm uses equation (5) as the fitness function. Equation (5) calculates the Information Content (IC) of the TFBSs as shown in Equation (2).

Equation (5) defines the length of subsequence W. The condition is $0 < W <= L$-1 and L is the length of the input sequence. The possible alphabets are b = {'A', 'C', 'G', 'T'}. The frequency of alphabet b appearing in the result of the particle is $f\,'_b$ calculated from equation (6) and the frequency of alphabet b not being in the results of particles is p'_b calculated from equation (7).

$$fitness = \sum_{i=1}^{W} IC \tag{5}$$

$$f'_b = \frac{c_b + d_b}{N - 1 + D} \tag{6}$$

$$p'_b = \frac{c_{0b} + db}{S + D} \tag{7}$$

The symbols and variables are described below:

- c_b is the number of times any alphabet b appears in the subsequences within each column.
- c_{0b} is the number of times any alphabet b appears outside the selected subsequences (background).
- N is the total number of input sequences.
- S is the total number of alphabets not selected within the chosen subsequences.
- d_b is the pseudo counts [2].
- D is the sum of pseudo counts.

H. Input data Collection and NexusPSO Algorithm

The Nexus algorithm is the pre-process consisting of: the grouping of subsequences (grouping), creation of connections between the subsequences (initializing), and the process of selecting the most suitable related pairs in the first two ranks (selection). This research collects relation tables, which consist of: table of input sequences, table of total possible subsequences, and table of particle data. PSO randomly selects subsequences from the Nexus procedure. The Pseudocode of the NexusPSO algorithm is as follow:

```
Algorithm NexusPSO
Input: $w$ = the length of subsequence, $Maximum$ = number of iterations, $N$ = number of input sequences, $L$ = length of input sequences, $b$ = {'A', 'C', 'G', 'T'}.
Output: the set of subsequences $CoM$
1: Nexus process (pre-process)
1.1: for $i$ = 1 to $N$ do
1.2:       for $j$ = 1 to $L_i-w+1$ do
1.3:         grouping $M[i][j]$;
1.4:         connection: $M[i][j]$ and $M[i+1][j]$;
           end for i
     end for i

2. PSO process
2.1. Initialize particles from best connection pair, start from first of sequences.
2.2. Particle movement
2.3      for $k$ = 1 to $Maximum$ OR not converged do
2.4          select local best particle;
2.5          update velocity of particles;
2.6          update position of particles;
2.7             if $k$ = 1 or local best > global best then
                    update global best from local best;
                end if
         end for $k$
```

IV. Experiment

A. Dataset and Parameter Settings

The dataset of genome sequences to be tested for efficiency and accuracy of NexusPSO algorithm consists of 3 groups as follows:

- *Saccharomyces Cerevisiae* [28] from the database SCPD. The length of input DNA sequences is 550 nucleotide pairs (550 alphabets) with other properties as shown in Table IV.
- *Homo sapiens* [29] from the database JASPAR. The length of input DNA sequences is 600 nucleotide pairs (600 alphabets) with other properties as shown in Table V.
- *Escherichia coli: E.Coli* [27] from the dataset of cyclic-AMP receptor protein (CRP) with properties as shown in Table VI. The length of each input DNA sequence is 105 nucleotide pairs (105 alphabets). The length of motif is defined to be 22 nucleotides [27]. This genome sequence has at least one TFBS sequence in each input DNA sequence. Also, these sequences have varied nucleotide patterns, which make them a popular data set to test the efficiency of detection algorithms [3, 5, 8, 11, 18, 22].

TABLE IV
PROPERTIES OF THE GENOME SEQUENCES OF *SACCHAROMYCES CEREVISIAE*

TF	Size	Length	Consensus Sequence
GAL4	6	17	CGGNNNNNNNNNNNCCG
RAP1	16	7	RMACCCA
REB1	14	7	YYACCCG
MCB	6	6	WCGCGW
PDR3	7	8	TCCGYGGA

TABLE V
PROPERTIES OF THE GENOME SEQUENCES OF *HOMO SAPIENS*

TF	Size	Length	Consensus Sequence
ELK4	20	9	ACCGGAAGT
E2F1	10	8	TTTGGCGC
FOXD1	20	8	GTAAACAT
USF1	30	7	CACGTGG
RELA	18	10	GGGAATTTCC

The parameter settings of particles in Nexus PSO algorithm are shown in Table VII which are proper data for the tested dataset [22].

B. Operation

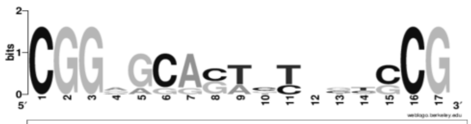
This research developed the NexusPSO algorithm using the C# language, version 5.0 in the Windows operating system. This research also used the SQL Server 2012 database management system as the design-related database in order to store the data of: DNA sequences, data of relations between all possible motifs, and the particle data. This research employs Weblogo (https:// weblogo.berkeley.edu/logo.cgi) to generate consensus sequences, which were used to analyze the efficiency of results gained from the NexusPSO algorithm.

TABLE VI
DATA OF TFBSS OF THE DATASET OF *ESCHERICHIA COLI*

No.	Names	Motif 1	Motif 2	No.	Names	Motif 1	Motif 2
1	CE1CG	17	61	10	ECOMALBA	14	
2	ECOARABOP	17	55	11	ECOMALBA2	61	
3	ECOBGLR1	76		12	ECOMALT	41	
4	ECOCRP	63		13	ECOOMPA	48	
5	ECOCYA	50		14	ECOTNAA	71	
6	ECODEOP2	7	60	15	ECOXUL	17	
7	ECOGALE	42		16	PBR-P4	53	
8	ECOILVBPR	39		17	TRN0CAT	1	84
9	ECOLAC	9	80	18	TDC	78	

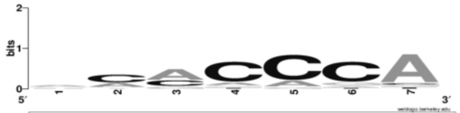
TABLE VII
PROPERTIES OF THE PARAMETERS FOR PARTICLES

Description	Parameters	Size
inertia weight	α	0.4
cognitive	β	0.8
social	γ	0.8
number of interations	*Maximum*	3000



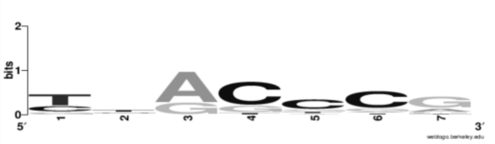
NexusPSO result					
1	CGGAGCACTGTTGAGCG	3	CGGAAGACTCTCCTCCG	5	CGGCGCACTCTCGCCCG
2	CGGAGCAGTGCGGCGCG	4	CGGGGCGGATCACTCCG	6	CGGGGCAGACTATTCCG

Fig. 4. Results of CS from the group of DNA sequences GAL4



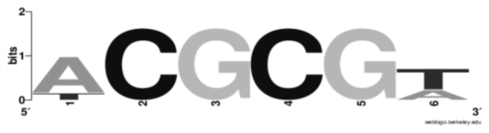
NexusPSO result					
1	ACACCCA	7	ACCCAGT	13	AAACCCA
2	GAACCCA	8	GAACCCA	14	CCGTGCA
3	ACACCCA	9	GCACCCA	15	GCACCCA
4	ACCCATG	10	TGCACCA	16	ACACCCA
5	GACACGC	11	CACCCAA		
6	CTGGACA	12	ACACCCA		

Fig. 5. Results of CS from the group of DNA sequences RAP1



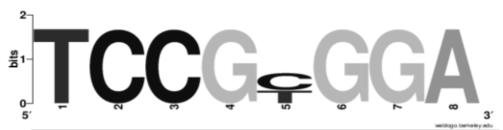
NexusPSO result					
1	TTACCCT	6	CCGGGCA	11	CTACCCG
2	TAACCCG	7	GGGTAGC	12	GTACGGG
3	TTACCCG	8	TCACCCG	13	CGGGTAA
4	TCACCCG	9	TTACCCG	14	TTACCCG
5	TTACCCG	10	CGGGTTA		

Fig. 6. Results of CS from the group of DNA sequences REB1



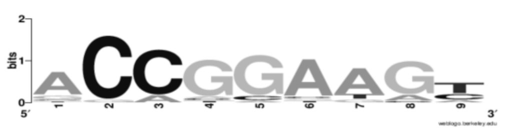
NexusPSO result					
1	ACGCGT	3	ACGCGT	5	ACGCGT
2	ACGCGA	4	TCGCGA	6	ACGCGT

Fig. 7. Results of CS from the group of DNA sequences MCB



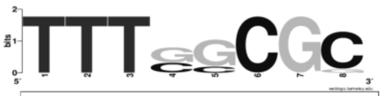
NexusPSO result					
1	TCCGCGGA	4	TCCGTGGA	6	TCCGTGGA
2	TCCGCGGA	5	TCCGTGGA	7	TCCGCGGA
3	TCCGCGGA				

Fig. 8. Results of CS from the group of DNA sequences PDR3



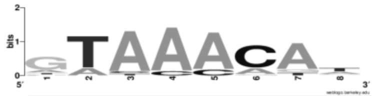
NexusPSO result							
1	ACCGGAAGT	6	ACCGGAAGT	11	ACCGGAAGC	16	GGATATagc
2	ACCGGAAGT	7	ACCGGATGT	12	acaACCGGA	17	ACCGGAAAC
3	ACCGGAAGT	8	ACCGGATGT	13	GCCGGAAGT	18	TCCGGAAGC
4	ACCGGAAGT	9	acACCGGAA	14	CCCGGAAGT	19	GCCGGAAAT
5	ACCGGAAGT	10	ACCGGAAGC	15	ACCGGATGC	20	ACCGGAACA

Fig. 9. Results of CS from the group of DNA sequences ELK4



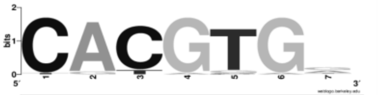
NexusPSO result			
1	TTTGGCGC	6	TTTCGCGC
2	TTTGGCGC	7	TTTCGCGC
3	TTTGGCGC	8	TTTGCCGC
4	TTTGGCGC	9	TTTCCCGC
5	TTTCGCGC	10	TTTGGCGG

Fig. 10. Results of CS from the group of DNA sequences E2F1



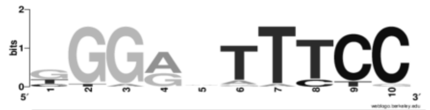
NexusPSO result							
1	GTAAACAT	6	GTAAACAT	11	TAAACAAt	16	ATAAACAA
2	GTAAACAT	7	GTAAACAT	12	GTAAACAA	17	CTAAACAG
3	GTAAACAT	8	GTAAACAA	13	AAACACgt	18	GTCAACAG
4	GTAAACAT	9	AAACAAtg	14	GTAAACAC	19	GTAACAAT
5	GTAAACAT	10	GTAAACAA	15	TAAACAGa	20	gTTAAGTA

Fig. 11. Results of CS from the group of DNA sequences FOXD1



NexusPSO result							
1	CACGTGG	9	CACGTGG	17	CACGTGA	25	CACGTGC
2	CACGTGG	10	CACGTGG	18	CACGTGA	26	TGTGGga
3	CACGTGG	11	CACGTGG	19	CGTGTaa	27	CATGTGA
4	CACGTGG	12	CACGTGG	20	CACGTGT	28	CACATGA
5	CACGTGG	13	CACGTGA	21	CACGTGT	29	CACGCGG
6	CACGTGG	14	CACGTGA	22	CACGTGT	30	CACGGGA
7	CACGTGG	15	CACGTGA	23	CACGTGT		
8	CACGTGG	16	CACGTGA	24	CACGTGC		

Fig. 12. Results of CS from the group of DNA sequences USF1



NexusPSO result							
1	GGGAATTTCC	6	GGAATTTCCg	11	CGGACTTTCC	16	tGGGGTTTTC
2	GGGAATTTCC	7	GGGGATTTCC	12	CGGACTTTCC	17	GGGGGATTCC
3	GGGAATTTCC	8	GGGACTTTCC	13	GGGGAATTCC	18	TGGGTTTCCc
4	TGGAATTTCC	9	CGGAGTTTCC	14	TGGGGTTTCC		
5	TGGAATTTCC	10	GGGAATTCCC	15	gtGGGGATTC		

Fig. 13. Results of CS from the group of DNA sequences RELA

C. *Experimental Results for Anal yzing the Efficiency*

The results from consensus sequences using the NexusPSO algorithm to detect the motif sequences in the genome sequences of *Saccharomyces cerevisiae* dataset are shown in Figures 4 to 8. The consensus sequences results for *Homo sapiens* are shown in Figures 9 to 13. Table IV and Table V show the consensus sequences of NexusPSO algorithm are identical to the consensus sequences from DNA footprinting methods.

Table VIII shows the representative sequences result of NexusPSO, selected by average IC value from all 18 runs, compared to the results from the traditional algorithms consisting of AlignACE [3], MEME [5], and Gibbs sampler [8] to detect the motifs in the genome sequences of *Escherichia coli*. Also, Table VIII compares the positions of motif sequences obtained from each algorithm with the positions of TFBSs. The Gibb Sampler results reveal 2 motif sequences with results more than 20 positions from TFBSs, the 5th DNA sequences (ECOYA) and the 17th DNA sequences (TRN9CAT). The AlignACE results show there are 2 motif sequences more than 15 positions from the TFBSs, the 7th DNA sequence (ECOGALE) and the 17th DNA sequence (TRN9CAT). Both algorithms do not have any resulting motif sequences match TFBSs. Results from the MEME algorithm show there are 4 motif sequences more than 20 positions from TFBSs and 1 motif sequence 16 positions from the TFBS, the 5th DNA sequences (ECOCYA), the 15th (ECOXUL), the 16th (PBR-P4), the 17th (TRN9CAT), and the 11th (ECOMALBA2), respectively, while 11 motif sequences match the TFBSs.

TABLE VIII
COMPARISON ON THE RESULTS OF TRADITIONAL ALGORITHMS, RELEVANT ALGORITHMS, AND NEXUSPSO ALGORITHM

		Traditional Algorithm						Related Work							
No.	BS	Gibbs Sampler	diff	Align ACE	diff	MEME	diff	GA	diff	PSO	diff	ACRI	diff	NexusPSO	diff
1	17,61	59	2	63	2	61	0	62	1	61	0	63	2	61	0
2	17,55	53	2	57	2	55	0	56	1	55	0	57	2	55	0
3	76	74	2	48	2	76	0	77	1	76	0	78	2	76	0
4	63	59	4	65	2	63	0	64	1	63	0	65	2	63	0
5	50	11	39	52	2	13	37	51	1	50	0	52	2	50	0
6	7,6	5	2	9	2	7	0	8	1	7	0	9	2	7	0
7	42	40	2	26	16	42	0	43	1	24	18	44	2	42	0
8	39	3	2	41	2	39	0	40	1	39	0	41	2	39	0
9	9,80	7	2	11	2	9	0	10	1	9	0	11	2	9	0
10	14	12	2	16	2	14	0	15	1	14	0	16	2	14	0
11	61	59	2	63	2	35	16	62	1	61	0	63	2	61	0
12	41	47	6	43	2	34	7	42	1	41	0	43	2	41	0
13	48	46	2	50	2	48	0	49	1	48	0	50	2	48	0
14	71	69	2	73	2	71	0	72	1	71	0	73	2	71	0
15	17	15	2	19	2	75	58	18	1	17	0	19	2	17	0
16	53	49	4	55	2	6	47	54	1	53	0	55	2	53	0
17	1,84	25	24	68	16	27	26	56	28	5	4	95	11	5	4
18	78	74	4	80	2	16	2	77	1	76	2	78	0	76	2

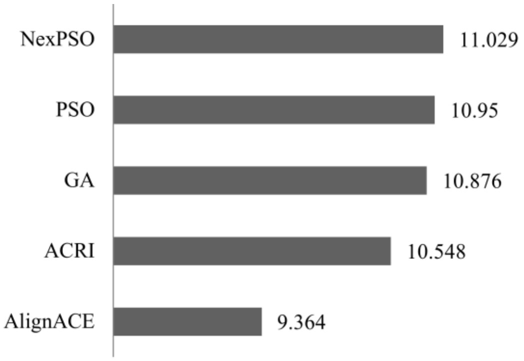


Fig. 14. Comparison results of IC value among AlignACE, GA, PSO, ACRI, and NexusPSO

According to the data in Table VIII, the motif sequence results of the GA [11] and ACRI [21] algorithms for the 17th DNA sequence (TRN9CAT) are shifted from the TFBS by 28 and 11 positions respectively. The result of the PSO [30] algorithm for the 7th sequence is also shifted from TFBS by 18 positions. This shows these algorithms cannot detect the motif sequences of the 17th and 7th DNA sequence (TRN9CAT, ECOGALE) accurately. The NexusPSO algorithm had the most accurate detection of the motif sequence in the 17th DNA sequence (TRN9CAT) with a deviation from the TFBS of only 4 positions. Also, NexusPSO detected the motif sequences by completely

TABLE IX
AVERAGE IC VALUES FROM 18 RUNS AMONG THE DIFFERENT ALGORITHMS

MEME	AlignACE	ACRI	NexusPSO
9.508	9.752	10.273	11.030

TABLE X
IC VALUES FROM 18 RUNS AMONG THE DIFFERENT ALGORITHMS

No.	MEME	AlignACE	ACRI	NexusPSO
1	10.032	9.651	10.01	11.045
2	9.075	9.887	10.28	11.804
3	10.02	9.576	9.987	11.018
4	10.05	9.624	10.403	10.946
5	9.117	10.235	10.457	10.354
6	9.892	9.71	10.184	11.934
7	9.554	9.01	9.895	11.005
8	10.124	9.934	10.258	11.112
9	9.646	9.807	10.354	10.124
10	9.439	9.853	10.421	11.053
11	9.121	10.12	10.53	10.984
12	9.16	9.399	10.415	10.852
13	9.684	9.976	10.38	11.074
14	9.773	9.825	10.286	11.04
15	9.024	9.769	10.179	11.206
16	9.008	10.314	10.3	11.704
17	9.105	9.011	10.14	11.029
18	9.32	9.835	10.431	10.254

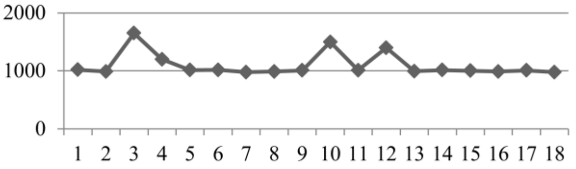


Fig. 15. The process times of NexusPSO for 18 runs

TABLE XI
T-VALUES OF NEXUSPSO COMPARED WITH OTHER ALGORITHMS

t-value	MEME	AlignACE	ACRI
compared with NexusPSO	10.354	9.181	6.34

matching the TFBSs for 16 sequences, resulting in the NexusPSO algorithm having the highest IC value, as shown in Fig. 14. Table IX shows the average IC values from 18 runs among the different algorithms including MEME, AlignACE, ACRI and NexusPSO. Table X shows the IC values of each run. The computational times are between 980 and 1650 milliseconds as shown in Fig. 15. The comparison of *t*-values among the relevant algorithms including NexusPSO is shown in Table XI. Considering *t*-test from 18 samples, the degree of freedom is 18+18-2 = 34 and let the significance level is α = 0.05 (confidence level is 95%), so that $t_{0.95}(34) = 1.691$. Comparing to *t*-value of NexusPSO from Table XI, we found that *t*-value of NexusPSO is higher than $t_{0.95}(34)$.

V. CONCLUSIONS

There are many algorithms available for detecting TFBSs, many of which were tested in this study. The Nexus procedure is designed to manage the problem space to become smaller, helping the random process of the algorithm avoid local optimums results.

The data from this study shows that NexusPSO can detect TFBSs more efficiently and accurately than other available methods. According to the samples in this study, NexusPSO have the highest IC at 11.029 scoring better than previously recorded results for PSO [30], GA [11] and ACRI [21] which had IC values of 10.95, 10.876, and 10.548, respectively. Considering *t*-test, it indicates that there are different significances between the information content by NexusPSO and other algorithms. Furthermore, the results of consensus sequences of NexusPSO show efficient results when compared to the results from DNA footprinting method.

However, the NexusPSO algorithm still needs to develop the competence to detect TFBSs with multiple motifs in each input sequence.

REFERENCES

[1] L. Elnitski, V.X. Jin, P.J. Farnham, S.J.M. Jones, "Locating mammalian transcription factor binding sites: a survey of computational and experimental techniques," *Genome Research*, vol.16, pp. 1455–1464, 2006.

[2] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, J.C. Wootton, "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignmzent," *Science*, vol. 262, pp. 208-214, 1993.

[3] J.D. Hughes, P.W. Estep, S. Tavazoie, G.M. Church, "Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae," *Journal of Molecular Biology*, vol. 296, pp. 1205–1214, 2000.

[4] X. Liu, D.L. Brutlag, J.S. Liu, "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes," *Pacific Symposium on Biocomputing*, 2001, pp. 127–138.

[5] T.L. Bailey, N. Williams, C. Misleh, W.W. Li, "MEME: discovering and analyzing DNA and protein sequence motifs," *Nucleic Acids Research*, vol. 34, pp. 369–373, 2006.

[6] G. Pavesi, P. Mereghetti, F. Zambelli, M. Stefani, G. Mauri, G. Pesole, "MoD Tools: regulatory motif discovery in nucleotide sequences from co-regulated or homologous genes," *Nucleic Acids Research*, vol. 34, pp. 566–570, 2006.

[7] X.S. Liu, D.L. Brutlag, J.S. Liu, "An algorithm for finding protein–DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments," *Nature Biotechnology*, vol. 20, pp. 835–839, 2002.

[8] A.F. Neuwald, J.S. Liu, C.E. Lawrence, "Gibbs motif sampling: detection of bacterial outer membrane protein repeats," *Protein Sci.*, vol. 4, no. 8, pp. 1618–1632, 2004.

[9] F.F.M. Liu, J.J.P. Tsai, R.M. Chen, S.N. Chen and S.H. Shih, "FMGA: finding motifs by genetic algorithm," *IEEE Fourth Symposium on Bioinformatics and Bioengineering (BIBE 2004)*, pp. 459–466, 2004.

[10] C. Notredame, D.G Higgins, "SAGA: Sequence alignment by genetic algorithm," *Nucleic Acids Res.*, vol. 24, no. 8, pp. 1515–1524, 1996.

[11] D. Che, Y. Song, K. Rasheed, "MDGA: motif discovery using a genetic algorithm," *Genetic and Evolutionary Computation (GECCO 2005)*, pp. 447–452, 2005.

[12] K. Socha, M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, 2008.

[13] D.D. Duc, H.Q. Dinh, H.H. Xuan, "On the pheromone update rules of ant colony optimization approaches for the job shop scheduling problem," *in: Proceedings of the 11th Pacific Rim International Conference on Multi-Agents, Intelligent Agents and Multi-Agent Systems*, vol. 5357, pp. 153–160, 2008.

[14] V. Maniezzo, A. Carbonaro, "An ANTS heuristic for the frequency assignment problem," *Future Gener. Comput. Syst*., vol. 16, no. 8, pp. 927–935, 2000.

[15] S.J. Shyu, C.Y. Tsai, "Finding the longest common subsequence for multiple biological sequences by ant colony optimization," Comput Oper Res, vol.36, no. 1, pp. 73-91, 2009.

[16] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evaluation and Applications 2008*, 2008, pp. 1–10.

[17] J. Kennedy, R. Eberhart, "Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks," *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[18] M. Dorigo, V. Maniezzo and A. Colomi, "Ant system: optimization by a colony of coorperating agents," *IEEE Trans. Syst. Man Cybern.—Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[19] J. Yan, M. Li, and J. Xu, "An Adaptive Strategy Applied to Memetic Algorithms," *IAENG International Journal of Computer Science*, vol. 42, no. 2, pp. 73-84, 2015.

[20] D. Zhu, L. Wang, J. Tian and X. Wang, "A Simple Polynomial Time Algorithm for the Generalized LCS Problem with Multiple Substring Exclusive Constraints," *IAENG International Journal of Computer Science*, vol. 42, no. 2, pp. 214-220, 2015.

[21] W. Liu, H. Chen, L. Chen, "ACRI: an ant colony optimization based algorithm for identifying gene regulatory elements," *Computer in Biology and Medicine*, vol. 43, pp. 922-932, 2013.

[22] M. Karabulut, T. Ibrikci, "PSO-variants: a Bayesian Scoring Scheme based Particle Swarm Optimization algorithm to identify transcription factor binding sites," *Applied Soft Computing*, vol. 12, pp. 2846-2855, 2012.

[23] Z. Wei, S.T. Jensen, "GAME: detecting cis-regulatory elements using a genetic algorithm," *Bioinformatics*, vol. 22, pp. 1577–1584, 2006.

[24] T.M. Chan, K.S. Leung, K.H. Lee, "TFBS identification based on genetic algorithm with combined representations and adaptive post-processing," *Bioinformatics*, vol. 24, pp. 341–349, 2008.

[25] M. A. H. Akhand, S. Akter, M. A. Rashid and S.B. Yaakob, "Velocity Tentative PSO: An Optimal Velocity Implementation based Particle Swarm Optimization to Solve Traveling Salesman Problem," *IAENG International Journal of Computer Science*, vol. 42, no. 2, pp. 221-232, 2015.

[26] J. Kennedy, R. Mendes, "Population structure and particle swarm performance, in: Proceedings of the Evolutionary Computation on 2002. CEC '02, Proceedings of the 2002 Congress – Vol. 02," *IEEE Computer Society*, vol. 02, pp. 1671–1676, 2002.

[27] G.D. Stormo, G.W. Hartzell, "Identifying protein-binding sites from unaligned DNA fragments," *Proc. Natl Acad. Sci.USA*, vol. 86, no. 4, pp. 1183–1187, 1989.

[28] J. Zhu, M.Q. Zhang, "SCPD: a promoter database of the yeast Saccharomyces cerevisiae," *Bioinformatics*, vol. 15, no. 7–8, pp. 607–611, 1999.

[29] J.C. Bryne, E. Valen, et al., "JASPAR:the open access database of transcription factor-binding profiles: new content and tools in the 2008 update," *Nucleic Acids Res.*, vol. 36, pp. 102–106, 2008.

[30] H. Ge, L. Sun, Y. Yao and J. Yu, "An automatic motif regnition algorithm in DNA sequences based on particle swarm optimization and random projection," *Bioinformatics and Biomedicine (BIBM)*, 2017, pp. 2241-2243.

Sarawoot Som-in received his B.S. in the Department of Computer Science from Huachiew Chalermprakiet University, Bangkok, Thailand in 2005. He received the Master degree in the Department of Computer Science from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand in 2010, where he is currently working toward the Ph.D. degree. His research interests include design and analysis of algorithm and bioinformatics.

Warangkhana Kimpan received her Ph.D. degree in System Information Engineering from Kagoshima University, Japan. She is currently an assistant professor in Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. Her main research interests are Swarm Intelligence, Biomedical Engineering, Big Data, Data Science and analytics, Cloud Computing, and Internet of Things.