Refining the Scoring Matrix for the Alignment of Intrinsically Disordered Proteins with Meta-heuristic Algorithms*

Tz-Hsu Lee[1], Kuo-Si Huang[2] and Chang-Biau Yang[1†]
[1]Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan
[†]cbyang@cse.nsysu.edu.tw
[2]Department of Business Computing
National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan

Abstract

Scoring matrices play an important role of protein sequence alignment in bioinformatics, where BLOSUM and PAM are two of the famous and widely used instances. The protein sequence alignment can be used for detecting homology, an essential analysis to newly determined sequences. However, applying common scoring matrices may get worse alignment results for protein sequences with rich disordered regions. In this paper, we focus on refining the scoring matrix based on machine learning by hybridizing the meta-heuristic algorithms, which are conceptually simpler and require less computation than the exhaustive search. Several famous meta-heuristic algorithms are considered for evaluating with the EUMAT dataset. According to the analysis of local and global search, we design the hybrid algorithm based on PSO+TLBO (particle swarm optimization, teaching learning based optimization) with the diversity-guided strategy. In the 10-fold cross-validation experiments with the whole EUMAT dataset, our algorithm gains coverage improvement about 6.6%, 4.8%, 1.8%, 6.6%, 5.6%, and 5.8% compared to BLOSUM, EDSSMat, GAHS, MDM, PAM, and VTML, respectively. By the t-test, these improvements are statistically significant.

Keywords: Protein Sequence Alignment, Scoring Matrix, Intrinsically Disordered Proteins, Meta-heuristic Algorithms.

1 Introduction

Because of the popularity of computers and information technologies, computer programs for bioinformatics become easier to implement, such as molecular evolution, homology modeling, and protein functions. In proteomics, the molecular structures of proteins, such as the three-dimensional (3-D) structures, can help us realize their biological functions and interactions between proteins and ligands. However, some proteins have unstable globular 3-D structures, called *intrinsically disordered proteins* (IDPs) [3, 6], due to a lack of fixed 3-D structures. Moreover, the IDPs may directly affect the accuracy of protein sequence alignments [21].

The variability of IDPs has become a fundamental issue in some biochemical reactions because they provide wider structure flexibilities. In addition, because of lacking a fixed structure, it is possible to combine with different proteins of various structures. When the IDPs combine to different proteins, they may change their shapes to interact with corresponding proteins [4, 5]. In homology detection, an appropriate protein sequence alignment will get more accurate identifications of homology.

There are two types of famous scoring matrices used for protein sequence alignment, the *blocks substitution matrix* (BLOSUM) [8] and the *point accepted mutation* (PAM) matrix [2]. But, these two matrices may not be suitable for disordered proteins because the frequencies of residue substitutions in the disordered regions are higher than the ordered regions [2, 8].

The *genetic algorithm harmony search* (GAHS), proposed by Tsai *et al.* [22] in 2021, is a hybrid *meta-heuristic algorithm* (MA) that takes the *genetic algorithm* (GA) as the basis to improve the homology detection in IDPs. In order to obtain a better scoring matrix for covering the disordered proteins, we take the *particle swarm optimization* (PSO) [7] and the *teaching learning based optimization* (TLBO) [18] as the bases, because their encoding mechanism of population is similar to biological mechanisms and the mathematical calculation is suitable for finding the scoring matrix. The EUMAT dataset provided by Trivedi and Nagarajaram [21] concerns the disordered proteins. To improve the scoring matrix, we try to adjust the crossover and mutation operators to make the stronger evolution for avoiding the local optimum by combining some outstanding MAs.

In this paper, we propose the hybrid algorithm of PSO+TLBO with diversity-guided, and then perform five times of 10-fold cross-validation for the entire EUMAT dataset. The experimental results show that our scoring matrix gets the highest average coverage,

*This research work was partially supported by the Ministry of Science and Technology of Taiwan under contract MOST 109-2221-E-110-040-MY2.

†Corresponding author.

66.58%, compared to other scoring matrices. It improves the average coverage about 6.6%, 4.8%, 1.8%, 6.6%, 5.6%, and 5.8% compared to BLOSUM [8], EDSSMat [21], GAHS [22], MDM [11], PAM [2], and VTML [16], respectively. Besides, we use the t-test to verify the significant difference between our scoring matrix and others.

The organization of this paper is given as follows. Section 2 introduces the alignment of protein sequences and homology detection in biology. Section 3 describes our method, and Section 4 shows experimental results. Finally, Section 5 summarizes this paper and gives some future works.

2 Protein Sequence Alignment

The purpose of homology detection is to classify whether the two species are originated from the same ancestor or not. The sequence alignment is one of the possible ways for homology detection.

Given two protein sequences $A = a_1a_2a_3\cdots a_m$ and $B = b_1b_2b_3\cdots b_n$, consisting of amino acids, the alignment score of A and B can be calculated by the following dynamic programming [20]:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + \delta(a_i, b_j), \\ \max_{1\le u_1\le i}\{H_{i-u_1,j} - \gamma_{u_1}\}, \\ \max_{1\le u_2\le j}\{H_{i,j-u_2} - \gamma_{u_2}\}, \\ 0. \end{cases} \quad (1)$$

In Eq. 1, $H_{i,j}$ denotes the alignment score of $A_{1..i}$ and $B_{1..j}$, $\delta(a_i, b_j)$ is the score for aligning a_i and b_j together, obtained from a *scoring matrix*, γ_{u_1} and γ_{u_2} are the gap penalties of lengths u_1 and u_2, respectively, where $1 \le i \le m$ and $1 \le j \le n$. One example of the scoring matrix is shown in Figure 1. If these two sequences are highly similar, they may be regarded to have the same evolutionary ancestor.

Some scoring matrices, such as PAM [2], BLOSUM [8], MDM [11], and VTML [16] are usually used for the alignment of protein sequences. However, there are many naturally existing functional proteins that contain unstable 3-D structures and appear to be unfolded. Such proteins are called *intrinsically disordered proteins* (IDPs) [3, 6]. The particular compositions of amino acids and the higher evolution rate in IDPs indicate that substitution frequencies of residues in disordered regions are different from that in ordered regions. Thus, the above scoring matrices may not be suitable for detecting homology in proteins enriched with disordered regions [21]. EDSSMat [21] are scoring matrices particularly designed for IDPs.

3 Our Method

This paper tries to build better scoring matrices for aligning IDPs. We take the EDSSMat scoring matrices [21] as the basis and try to refine it. In our method, we transform a scoring matrix into a chromosome as shown

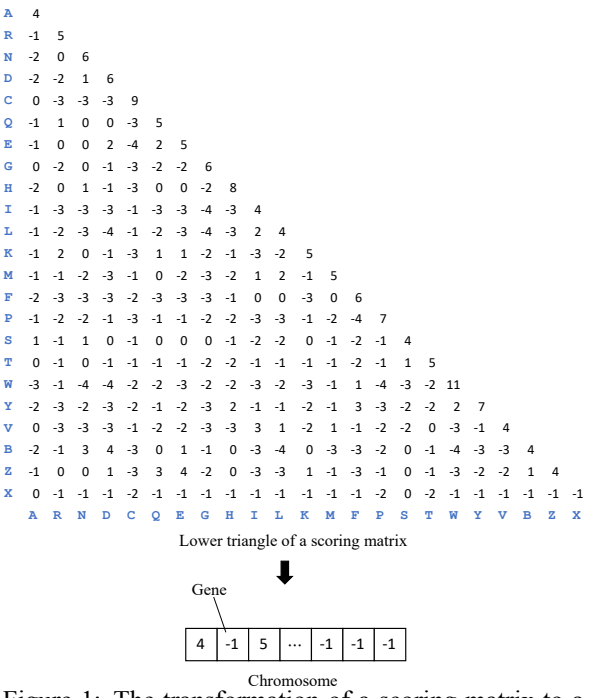

Figure 1: The transformation of a scoring matrix to a chromosome.

in Figure 1. The population consists of 27 chromosomes (scoring matrices). For initialization, 7 chromosomes are EDSSMat matrices and the other 20 chromosomes are generated randomly. We choose ten of metaheuristic algorithms, as shown in Table 1, and combine them to perform a primitive experiment.

In the primitive experiment, the PSO [7], TLBO [18], and PSO+TLBO with interleaving(PSO) get good performances, but they lack a strong mechanism to get out of the local optimum. Note that the PSO+TLBO with interleaving(PSO) is a hybrid algorithm of PSO and TLBO, where the type of hybrid is interleaving, which means that it performs two algorithms in an iteration, and the augment algorithm in interleaving(PSO) is performed first. Hence, we integrate the diversity-guided mechanism to avoid the local optimum. In the PSO+TLBO with diversity-guided (PTd), we first implement the PSO to evolve the population, and then invoke the TLBO to evolve the population produced by the PSO. In the end of PSO+TLBO with interleaving(PSO), we integrate the diversity-guided mechanism to decide the necessity of mutation.

Algorithm PTd: PSO+TLBO with diversity-guided.

Input: P: population size; d: dimension of problem; c_1: group-learning weight; c_2: self-learning weight; w: inertia weight; $v_{\min}$: minimum of velocity; $v_{\max}$: maximum of velocity; T_{D}: diversity threshold; G: maximal number of iterations.

Output: The scoring matrix with best coverage so far.

Step 1 (Preserve self-best and group-best): We preserve the matrix with the best fitness of every chro-

Table 1: The ten meta-heuristic algorithms (MAs) used in this paper. P: population size; G: maximal number of iterations.

Author(s)	Year	Algorithm	Parameters
Holland [9]	1975	Genetic algorithm (GA)	P, G, l, R_{cr}, R_{mu}
Eberhart and Kennedy [7]	1995	Particle swarm optimization algorithm (PSO)	P, G, c_1, c_2, w, v
Karaboga and Basturk [12]	2007	Artificial bee colony algorithm (ABC)	P, G, N_{ABC}, $N_{\text{ABC}}^{\text{limit}}$
Rashedi *et al.* [19]	2009	Gravitational search algorithm (GSA)	P, G, G_0, ε
Rao *et al.* [18]	2011	Teaching learning based optimization (TLBO)	P, G, F_{TLBO}
Yang [23]	2012	Flower pollination algorithm (FPA)	P, G, p_{FPA}
Rao and Patel [17]	2013	Improved teaching learning based optimization (ITLBO)	P, G, F_{ITLBO}
Mirjalili *et al.* [15]	2014	Grey wolf optimization algorithm (GWO)	P, G, a_{GWO}
Mirjalili [13]	2016	Sine Cosine algorithm (SCA)	P, G, a_{SCA}, r_1, r_2
Mirjalili and Lewis [14]	2016	Whale optimization algorithm (WOA)	P, G, a_{WOA}, b_{WOA}

mosome individually, as well as the matrix with the best fitness of all chromosomes.

Step 2 (Evolution of PSO): Evolve the population with Eqs. 2 and 3.

$$v_i(t+1) = w \times v_i(t) + c_1 \times \text{rand}[0,1] \times [x_i^*(t) - x_i(t)] + c_2 \times \text{rand}[0,1] \times [x^*(t) - x_i(t)], \quad (2)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (3)$$

where $x_i^*(t)$ is the matrix with the best fitness of chromosome i. Here, c_1 and c_2 are group-learning and self-learning weights to decide learning from the group or itself, respectively; and w is the inertia weight to decide the search direction to global or local. w is decreased linearly as given in Eq. 4, so that the algorithm have both searching directions.

$$w = \frac{w_{\text{start}} - (w_{\text{start}} - w_{\text{end}})}{G \times t}, \quad (4)$$

where w_{start} and w_{end} are weight values of the initial and the end, respectively.

Step 3 (TLBO teaching): Every chromosome tries to move toward the teacher chromosome whose fitness is the best as given in Eq. 5.

$$x_i(t+1) = x_i(t) + \text{rand}[0,1] \times [x_{\text{T}}(t) - F_{\text{TLBO}} \times \overline{x}], \quad (5)$$

where $x_{\text{T}}(t)$ is the teacher chromosome with the best fitness of all chromosomes, $\overline{x}(t)$ is the mean chromosome where each gene represents the average of corresponding genes of this population, and F_{TLBO} is a teaching factor to decide the influence of $\overline{x}(t)$ as given in Eq. 6.

$$F_{\text{TLBO}} = \text{round}[1 + \text{rand}[0,1]]. \quad (6)$$

Step 4 (TLBO learning): After teaching by the teacher, chromosomes also learn with each other to improve their fitness by Eq. 7, where x_{rand} is a student chosen randomly.

$$x_i(t+1) = x_i(t) + \text{rand}[0,1] \times |x_i(t) - x_{\text{rand}}(t)|. \quad (7)$$

Step 5 (Diversity-guided mutation): The diversity (D) of population can reflect the degree of convergence as given in Eq. 8. As the diversity is less than threshold T_{D}, we perform mutations to make the population get out of the local optimum. The mutation rate r_{mu} is adaptive as given in Eq. 9. Moreover, we use the multiple-point mutation. It decides the number of mutated genes and corresponding mutation positions randomly, and it has a probability p_h to slightly adjust values of genes by ± 1.

$$D(t) = \frac{1}{P \times L_{\text{diag}}} \sum_{i=1}^{P} \sqrt{\sum_{j=1}^{d} [x_i^j(t) - \overline{x}^j(t)]^2}, \quad (8)$$

where d is the size of the scoring matrix; $L_{\text{diag}} = \sqrt{d} \times |u_{\text{max}} - u_{\text{min}}|$ denotes the maximal length of diagonal in the search space, if the boundary of every dimension is in $[u_{\text{min}}, u_{\text{max}}]$; and $\overline{x}^j(t) = (1/P) \sum_{i=1}^{P} x_i^j(t)$ for dimension j.

$$r_{\text{mu}}(i) = \begin{cases} R_2, & \text{if } f_i < f_{\text{avg}}; \\ R_2 \times \frac{f_{\text{max}} - f_i}{f_{\text{max}} - f_{\text{avg}}}, & \text{if } f_i \geq f_{\text{avg}}; \end{cases} \quad (9)$$

where $0 < R_2 < 1$ is the predefined mutation rate; f_i is the fitness of chromosome i; f_{max} and f_{avg} are the maximal and average fitness values of the chromosomes in iteration t, respectively.

Step 6 (Termination): In the end of PTd, there are 27 chromosomes. If it meets the termination condition, the algorithm terminates; otherwise, go to Step 1.

4 Experimental Results

The EUMAT dataset, provided by Trivedi and Nagarajaram [21], contains 36498 sequences of proteins. These protein sequences in the EUMAT dataset were retrieved from the UniProtKB [1]. In order to detect homologies with varying degrees of disorder, the EUMAT dataset is divided into three different datasets, including the less disordered (LD), moderately disordered (MD) and highly disordered (HD), corresponding to disorder percentage ranges $[0\%, 20\%)$, $[20\%, 40\%)$ and $[40\%, 100\%]$, respectively. The query sequences of

Table 2: The distribution of proteins in the EUMAT dataset [21].

Sub-dataset	Disorder percentage	Number of protein sequences	Number of protein families
LD	0% to 20%	27832	3352
MD	20% to 40%	5029	1460
HD	$> 40\%$	3637	938
Total	–	36498	5750

every experiment are all contained in the HD dataset. Table 2 shows the distribution of proteins in the EUMAT dataset.

Based on the primitive experiments, this paper proposes the PSO+TLBO with diversity-guided (PTd) for $c_1 = 2$, $c_2 = 1$, $w = 0.9$ and $F_{\text{TLBO}} \in \{1, 2\}$. We use the PTd to refine the scoring matrices for the whole EUMAT dataset and compare the average coverages and standard deviations to other widely used scoring matrices. We perform the 10-fold cross-validation randomly for 5 times with the whole EUMAT dataset. In each group of 10-fold cross-validation, we choose 9 folds for training, and the remaining fold for testing. So, we get 50 scoring matrices from the 10-fold cross-validation for 5 times. When the difference of coverage increase for 5 consecutive iterations does not exceed 0.01, the program terminates; otherwise, it executes until the maximal number of iterations is reached. The experiments are performed on a computer with Windows 10 64-bit OS, 3.60 GHz Intel(R) Core(TM) i7-4790 CPU and RAM of 12 GB. The algorithms are implemented with Python of version 3.8.5. The execution time of all training process with the whole EUMAT dataset is about 47 days.

For each scoring matrix obtained from one of 10-fold cross-validation, we also perform it on the 50 testing folds (10-fold for 5 times). Table 3 shows the average coverages and standard deviations of 50 scoring matrices. As we can see, the highest and the lowest average coverage is PTd1-5 with 66.58% and PTd4-9 with 60.21%, respectively. We also perform BLOSUM, EDSSMat, GAHS, MDM, PAM, and VTML on the 50 testing folds and compare with PTd1-5, as shown in Table 4. In fact, there are some other scoring matrices for disordered proteins [21], such as DUNMat, Disorder and MidicMat. We do not involve these scoring matrices in the performance comparison for our experiments, because Trivedi and Nagarajaram [21] showed that their EDSSMat is superior to these matrices with respect to the coverage measurement. Note that the GAHS scoring matrix is also trained by the EUMAT dataset in 10-fold cross-validation for 5 times, but its dividing way is different to ours. As shown in Table 4, PTD1-5 outperforms other matrices. These average coverages of PTD1-5 are 66.65%, 66.42%, 67.52%, 65.22% and 67.10% for the 5 times, and the total average coverage is 66.58%, which is obtained by averaging the 50 testing folds.

Note that the coverage obtained from the primitive experiment (with the small datasets) is higher 80%, however the coverage obtained from the 10-fold experiment (with the whole EUMAT dataset) is below 70%. This phenomenon is due to that a false positive result is more likely produced in a large dataset.

Finally, we use the paired two-sample t-test to determine whether the difference is significant between scoring matrices at a confidence level of 95% by calculating its t-value, as given in Eq. 10 [10].

$$t_{\text{test}} = \frac{\overline{Q_{\text{diff}}}}{\sqrt{\frac{\sigma^2_{\text{diff}}}{N_{\text{test}}}}}, \tag{10}$$

where $\overline{Q_{\text{diff}}}$ and σ^2_{diff} are the mean and the variance of coverage differences, respectively; N_{test} is the number of observations. We regard $t \geq 2.0096$ as significant corresponding to the 95% confidence.

The result of t-test is shown in Table 5. According to Table 5, we can see that our coverage improvements are totally statistically significant compared with other scoring matrices with t-test.

5 Conclusion

In this paper, we design a machine learning algorithm, combined by meta-heuristic methods, to refine the scoring matrix of sequence alignment for intrinsically disordered proteins. Through the analysis on primitive experiments, we propose the algorithm based on PSO+TLBO with diversity-guided strategy (PTd). For the EUMAT dataset, the experimental results show that the average coverage of the refined scoring matrix, obtained by PTd, improves about 6.6%, 4.8%, 1.8%, 6.6%, 5.6%, and 5.8% compared to BLOSUM, EDSSMat, GAHS, MDM, PAM, and VTML, respectively. These improvements are statistically significant by the t-test.

According to the experimental results, the meta-heuristic algorithms can refine and get better scoring matrices than the EDSSMat. But, it gets into a bottleneck of requiring a large amount of computational time on the sequence alignment. So, reducing the computational time of sequence alignment is a crucial issue that needs to be overcome for improvement in the future.

References

[1] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, P. Bansal, A. J. Bridge, S. Poux, L. Bougueleret, and I. Xenarios, *UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: how to use the entry view*. Springer, 2016.

[2] M. Dayhoff, R. Schwartz, and B. Orcutt, "A model of evolutionary change in proteins," *Atlas of Protein Sequence and Structure*, Vol. 5, pp. 345–352, 1978.

[3] A. K. Dunker, C. J. Brown, J. D. Lawson, L. M. Iakoucheva, and Z. Obradović, "Intrinsic disorder and protein function," *Biochemistry*, Vol. 41, No. 21, pp. 6573–6582, 2002.

Table 3: The average coverages and standard deviations of PTd scoring matrices for $c_1 = 2$, $c_2 = 1$, $w = 0.9$ and $F_{\text{TLBO}} \in \{1, 2\}$ in 50 testing folds.

Matrix	Highly disordered		Matrix	Highly disordered	
	Gap open & gap extension	Average coverage		Gap open & gap extension	Average coverage
PTd1-1	−17 & −1	0.6414 ± 0.06	PTd3-6	−12 & −1	0.6539 ± 0.06
PTd1-2	−19 & −2	0.6290 ± 0.05	PTd3-7	−18 & −2	0.6480 ± 0.06
PTd1-3	−17 & −1	0.6366 ± 0.07	PTd3-8	−10 & −2	0.6486 ± 0.06
PTd1-4	−14 & −3	0.6373 ± 0.06	PTd3-9	−16 & −2	0.6313 ± 0.06
PTd1-5	**−13 & −1**	**0.6658 ± 0.06**	PTd3-10	−18 & −2	0.6520 ± 0.06
PTd1-6	−20 & −1	0.6463 ± 0.06	PTd4-1	−18 & −2	0.6363 ± 0.07
PTd1-7	−10 & −1	0.6527 ± 0.07	PTd4-2	−18 & −2	0.6268 ± 0.06
PTd1-8	−18 & −2	0.6447 ± 0.07	PTd4-3	−18 & −2	0.6206 ± 0.08
PTd1-9	−19 & −2	0.6513 ± 0.07	PTd4-4	−10 & −1	0.6305 ± 0.07
PTd1-10	−15 & −2	0.6353 ± 0.06	PTd4-5	−18 & −2	0.6118 ± 0.07
PTd2-1	−20 & −3	0.6442 ± 0.07	PTd4-6	−14 & −3	0.6248 ± 0.07
PTd2-2	−15 & −1	0.6354 ± 0.07	PTd4-7	**−18** & −2	0.6456 ± 0.07
PTd2-3	−18 & −2	0.6445 ± 0.06	PTd4-8	−19 & −2	0.6381 ± 0.06
PTd2-4	−18 & −2	0.6416 ± 0.06	**PTd4-9**	**−19 & −2**	**0.6021 ± 0.08**
PTd2-5	−14 & −1	0.6378 ± 0.07	PTd4-10	−18 & −2	0.6072 ± 0.07
PTd2-6	−20 & −3	0.6488 ± 0.05	PTd5-1	−13 & −3	0.6458 ± 0.07
PTd2-7	−15 & −1	0.6283 ± 0.07	PTd5-2	−19 & −2	0.6343 ± 0.06
PTd2-8	−18 & −2	0.6292 ± 0.06	PTd5-3	−10 & −2	0.6408 ± 0.05
PTd2-9	−18 & −2	0.6454 ± 0.07	PTd5-4	−20 & −2	0.6389 ± 0.06
PTd2-10	−11 & −1	0.6595 ± 0.06	PTd5-5	−17 & −1	0.6510 ± 0.06
PTd3-1	−13 & −2	0.6249 ± 0.07	PTd5-6	−14 & −3	0.6249 ± 0.07
PTd3-2	−15 & −1	0.6356 ± 0.06	PTd5-7	−10 & −3	0.6375 ± 0.06
PTd3-3	−19 & −1	0.6306 ± 0.07	PTd5-8	−17 & −1	0.6413 ± 0.07
PTd3-4	−17 & −1	0.6376 ± 0.06	PTd5-9	−14 & −2	0.6447 ± 0.06
PTd3-5	−18 & −2	0.6358 ± 0.06	PTd5-10	−12 & −3	0.6251 ± 0.06

Table 4: The average coverages and standard deviations of various scoring matrices in 10-fold cross-validation for 5 different times; the red mark represents the best coverage among the same group of scoring matrices.

Matrix	Gap open & gap extension	First time average coverage	Second time average coverage	Third time average coverage	Fourth time average coverage	Fifth time average coverage	Total average coverage
BLOSUM30	−18 & −3	0.5998 ± 0.07	0.5861 ± 0.06	0.5852 ± 0.09	0.5924 ± 0.07	**0.6050 ± 0.07**	0.5937 ± 0.07
BLOSUM50	−11 & −2	0.6149 ± 0.07	**0.6082 ± 0.06**	0.5852 ± 0.11	0.5928 ± 0.09	0.6036 ± 0.08	0.5970 ± 0.07
BLOSUM62	**−14 & −3**	0.6019 ± 0.07	0.5931 ± 0.08	**0.5959 ± 0.08**	0.5897 ± 0.07	0.6043 ± 0.08	**0.6001 ± 0.08**
BLOSUM80	−10 & −3	**0.6166 ± 0.08**	0.6055 ± 0.07	0.5813 ± 0.11	**0.5985 ± 0.09**	0.5886 ± 0.08	0.5981 ± 0.08
EDSSMat50	−18 & −2	0.6230 ± 0.08	0.6188 ± 0.07	0.6120 ± 0.07	0.6066 ± 0.07	**0.6288 ± 0.08**	0.6178 ± 0.07
EDSSMat60	−14 & −3	0.6259 ± 0.07	0.6123 ± 0.08	0.6114 ± 0.07	0.6023 ± 0.07	0.6270 ± 0.09	0.6158 ± 0.07
EDSSMat62	−19 & −2	0.6224 ± 0.08	0.6170 ± 0.07	0.6109 ± 0.07	0.6044 ± 0.07	0.6262 ± 0.08	0.6162 ± 0.07
EDSSMat70	−19 & −2	0.6234 ± 0.08	0.6174 ± 0.07	0.6099 ± 0.07	0.6037 ± 0.07	0.6270 ± 0.08	0.6163 ± 0.07
EDSSMat75	**−19 & −2**	0.6209 ± 0.08	**0.6229 ± 0.07**	**0.6269 ± 0.06**	0.5961 ± 0.08	0.6145 ± 0.08	**0.6179 ± 0.07**
EDSSMat80	−15 & −3	0.6265 ± 0.07	0.6214 ± 0.08	0.6077 ± 0.07	**0.6080 ± 0.07**	0.6143 ± 0.08	0.6175 ± 0.07
EDSSMat90	−19 & −2	**0.6272 ± 0.07**	0.6174 ± 0.07	0.6109 ± 0.07	0.6039 ± 0.07	0.6272 ± 0.08	0.6173 ± 0.07
GAHS77	**−17 & −2**	**0.6548 ± 0.07**	**0.6509 ± 0.04**	**0.6422 ± 0.08**	**0.6390 ± 0.07**	**0.6542 ± 0.07**	**0.6482 ± 0.06**
MDM10	−18 & −3	0.5673 ± 0.08	0.5567 ± 0.07	0.5636 ± 0.08	0.5514 ± 0.07	0.5419 ± 0.09	0.5562 ± 0.08
MDM20	−20 & −1	0.5820 ± 0.09	0.5746 ± 0.08	0.5878 ± 0.08	0.5791 ± 0.08	0.5652 ± 0.09	0.5778 ± 0.08
MDM40	**−20 & −3**	**0.6047 ± 0.08**	**0.6002 ± 0.07**	**0.6015 ± 0.08**	**0.5932 ± 0.07**	**0.5975 ± 0.08**	**0.5994 ± 0.07**
PAM120	**−7 & −1**	**0.6228 ± 0.07**	**0.6174 ± 0.07**	0.5854 ± 0.11	**0.6025 ± 0.10**	**0.6187 ± 0.07**	**0.6094 ± 0.08**
PAM250	−19 & −3	0.5982 ± 0.07	0.5851 ± 0.05	**0.5861 ± 0.07**	0.5854 ± 0.06	0.6069 ± 0.07	0.5923 ± 0.06
VTML10	−8 & −1	0.5557 ± 0.09	0.5462 ± 0.07	0.5594 ± 0.09	0.5539 ± 0.07	0.5348 ± 0.10	0.5500 ± 0.08
VTML20	−13 & −2	0.5752 ± 0.08	0.5635 ± 0.07	0.5768 ± 0.08	0.5752 ± 0.08	0.5706 ± 0.09	0.5714 ± 0.08
VTML40	−18 & −3	0.6001 ± 0.08	0.5862 ± 0.08	0.5827 ± 0.07	0.5791 ± 0.08	0.5914 ± 0.09	0.5879 ± 0.08
VTML80	−17 & −3	0.6098 ± 0.08	0.5961 ± 0.08	0.5973 ± 0.07	0.5876 ± 0.08	0.6092 ± 0.10	0.6000 ± 0.08
VTML120	**−13 & −3**	**0.6125 ± 0.07**	**0.6108 ± 0.07**	**0.6016 ± 0.08**	**0.5970 ± 0.07**	0.6156 ± 0.08	**0.6075 ± 0.07**
VTML160	−11 & −2	0.6110 ± 0.07	0.6003 ± 0.06	0.5813 ± 0.10	0.5870 ± 0.09	0.6135 ± 0.08	0.5986 ± 0.08
VTML200	−9 & −3	0.6116 ± 0.06	0.6014 ± 0.05	0.5820 ± 0.09	0.5922 ± 0.09	**0.6176 ± 0.06**	0.6010 ± 0.07
PTd1-5	**−13 & −1**	**0.6665 ± 0.06**	**0.6642 ± 0.05**	**0.6752 ± 0.07**	**0.6522 ± 0.06**	**0.6710 ± 0.08**	**0.6658 ± 0.06**

Table 5: The t-test for the coverage difference of PTd1-5 and other scoring matrices on 50 testing folds. t-value > 2.0096 and p-value < 0.05 correspond to the 95%-confidence.

Competitor	PTd1-5	
	t-value	p-value (two-tailed)
BLOSUM30	12.853	2.59e-17
BLOSUM50	7.901	2.70e-10
BLOSUM62	11.984	3.55e-16
BLOSUM80	7.502	1.11e-09
EDSSMat50	8.713	1.59e-11
EDSSMat60	9.108	4.08e-12
EDSSMat62	8.841	1.02e-11
EDSSMat70	8.981	6.28e-12
EDSSMat75	8.784	1.24e-11
EDSSMat80	8.911	7.99e-12
EDSSMat90	8.685	1.75e-11
GASH77	2.904	5.51e-03
MDM10	17.109	2.54e-22
MDM20	12.146	2.16e-16
MDM40	10.718	1.91e-14
PAM120	6.156	1.34e-07
PAM250	12.196	1.86e-16
VTML10	14.313	3.95e-19
VTML20	16.309	1.96e-21
VTML40	13.363	5.83e-18
VTML80	11.420	2.05e-15
VTML120	10.626	2.58e-14
VTML160	7.835	3.42e-10
VTML200	8.570	2.61e-11

[4] A. K. Dunker, J. D. Lawson, C. J. Brown, R. M. Williams, P. Romero, J. S. Oh, C. J. Oldfield, A. M. Campen, C. M. Ratliff, K. W. Hipps, J. Ausio, M. S. Nissen, R. Reeves, C. Kang, C. R. Kissinger, R. W. Bailey, M. D. Griswold, W. Chiu, E. C. Garner, and Z. Obradovic, "Intrinsically disordered protein," *Journal of Molecular Graphics and Modelling*, Vol. 19, No. 1, pp. 26–59, 2001.

[5] A. K. Dunker, P. Romero, Z. Obradovic, E. C. Garner, and C. J. Brown, "Intrinsic protein disorder in complete genomes," *Genome Informatics*, Vol. 11, pp. 161–171, 2000.

[6] H. J. Dyson and P. E. Wright, "Intrinsically unstructured proteins and their functions," *Nature Reviews Molecular Cell Biology*, Vol. 6, No. 3, pp. 197–208, 2005.

[7] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*, Nagoya, Japan, pp. 39–43, 1995.

[8] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proceedings of the National Academy of Sciences*, Vol. 89, No. 22, pp. 10915–10919, 1992.

[9] J. Holland, *Control and Artificial Intelligence*. University of Michigan Press, USA, 1975.

[10] H. Hsu and P. A. Lachenbruch, "Paired t test," *Encyclopedia of Biostatistics*, Vol. 6, pp. 1–2, 2005.

[11] D. T. Jones, W. R. Taylor, and J. M. Thornton, "The rapid generation of mutation data matrices from protein sequences," *Bioinformatics*, Vol. 8, No. 3, pp. 275–282, 1992.

[12] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," *Proceedings of the International Fuzzy Systems Association World Congress*, Cancun, Mexico, pp. 789–798, 2007.

[13] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-based Systems*, Vol. 96, pp. 120–133, 2016.

[14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, Vol. 95, pp. 51–67, 2016.

[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, Vol. 69, pp. 46–61, 2014.

[16] T. Müller, R. Spang, and M. Vingron, "Estimating amino acid substitution models: a comparison of dayhoff's estimator, the resolvent approach and a maximum likelihood method," *Molecular biology and evolution*, Vol. 19, No. 1, pp. 8–13, 2002.

[17] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, Vol. 20, No. 3, pp. 710–720, 2013.

[18] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, Vol. 43, No. 3, pp. 303–315, 2011.

[19] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, Vol. 179, No. 13, pp. 2232–2248, 2009.

[20] T. F. Smith and M. S. Waterman, "Comparison of biosequences," *Advances in Applied Mathematics*, Vol. 2, No. 4, pp. 482–489, 1981.

[21] R. Trivedi and H. A. Nagarajaram, "Amino acid substitution scoring matrices specific to intrinsically disordered regions in proteins," *Scientific Reports*, Vol. 9, No. 1, pp. 1–12, 2019.

[22] F.-Y. Tsai, C.-B. Yang, and K.-S. Huang, "Reconstructing the amino acid scoring matrices to improve homology detection in intrinsically disordered proteins," *Proceeding of Symposium on Digital Life Technologies (DLT2021)*, 8 pages, Pingtung, Taiwan, 2021.

[23] X.-S. Yang, "Flower pollination algorithm for global optimization," *Proceeding of the 11th International Conference on Unconventional Computing and Natural Computation*, Orléan, France, pp. 240–249, 2012.