RECONSTRUCTING THE AMINO ACID SCORING MATRIX TO IMPROVE HOMOLOGY DETECTION IN INTRINSICALLY DISORDERED PROTEINS*

[1]*Feng-Yang Tsai*, [1†]*Chang-Biau Yang*, and [2]*Kuo-Si Huang*

[1]Dept. of Computer Science and Engineering, National Sun Yat-sen University
[†]Corresponding Author, E-mail: cbyang@cse.nsysu.edu.tw
[2]Dept. of Business Computing, National Kaohsiung University of Science and Technology

ABSTRACT

In bioinformatics, the scoring matrices PAM and BLOSUM are famous and widely used for protein sequence alignment. They may not be always suitable for homology detection, such as in protein sequences with rich disordered regions. Unlike ordered regions, disordered regions have significant deviations of amino acid composition. Hence different amino acid scoring systems should be used for the ordered and disordered regions separately. This paper tries to reconstruct the scoring matrix by using the hybrid of the genetic algorithm and the harmony search. For evaluating the performance of the proposed algorithm, the EUMAT dataset is considered for the experiment of 10-fold cross-validation. The experimental results show that the average coverage 65.2% of the scoring matrix constructed by our algorithm, which improves the EDSSMat, BLOSUM, PAM, VTML, and MDM scoring matrices with 3.9%, 6.0%, 4.4%, 4.5%, and 5.0%, respectively.

Keywords: Homology Detection; Intrinsically Disordered Proteins; Sequence Alignment Scoring Matrix; Genetic Algorithm; Harmony Search.

1. INTRODUCTION

The three-dimensional structures of proteins can help us understand their biological functions, such as interactions with other proteins and ligands for drug design. But some intact proteins or protein regions do not spontaneously fold into well-organized globular structures, called *intrinsically disordered* [1]. Because these proteins lack a tightly ordered three-dimensional structure, they may have many different complex functions [1, 2]. Hence, accurate sequence alignments of disordered proteins [3] are needed in the studies of molecular evolution, homology modeling, protein functions, and so on. According to the comparative analysis of the characteristics of disordered proteins, such as sequence complexity, amino acid composition, and evolution rate, the evolutions of disordered regions plays important roles[4–8]. The evolution rates in intrinsically disorder regions are significantly higher than that in ordered regions. For example, more insertions and deletions may occur in disorder regions [9].

In molecular evolution, incorrect sequence alignments may lead to reconstruct erroneous phylogenetic trees. It may ultimately lead to erroneous analysis of the evolutionary relationships. In homology detection, if the three-dimensional structure of a protein is unknown, a more accurate sequence alignment may provide more accurate identifications of structural motifs. Better identifications of structural motifs may also help function predictions for unknown proteins. Two main scoring matrices are used for sequence alignment, the *blocks substitution matrix* (BLOSUM) [10] and the *point accepted mutation* (PAM) matrix [11]. The BLOSUM matrix set considers the blocks of highly conserved regions, while the PAM matrix is built from an overall alignment of highly similar and closely related proteins. The scores within each matrix are calculated by using logarithmic scales. Both BLOSUM and PAM scoring matrices are built with a variety of protein families [10, 11]. Therefore, the BLOSUM or PAM matrices may be suitable for the general proteins, but they may not be suitable for the disordered proteins.

Compared with the ordered region, the unique component bias and higher evolution rate of intrinsically disorder proteins show that the frequencies of residue substitutions in the disordered region are higher than the ordered region [12]. Therefore, it is not appropriate to use the scoring matrix built from the ordered re-

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

gions of the protein for performing sequence analysis in disordered regions, such as the *intrinsically disordered protein* (IDP) homology search. Hence, it is better to develop more suitable scoring matrices for disordered regions in proteins. Based on the EUMAT sequence dataset[3], this paper proposes a hybrid GAHS method, combining the *genetic algorithm* (GA)[13], and *harmony search* (HS)[14], to reconstruct the scoring matrix for protein sequence alignment. The experimental results show that both GA and HS can converge in this problem. Based on this observation, in our hybrid method, the GA can quickly obtain a good solution and the HS can reduce the solution space.

This paper focuses on producing a better scoring matrix for the disordered proteins, and we perform an experiment with 10-fold cross validation for 5 times. The experimental results show that our GAHS method obtains an average coverage 65.2%, while the highest average coverages of EDSSMat[3] is 61.3%. Our average coverage is also higher than the commonly used scoring matrices BLOSUM (59.2%), PAM (60.8%), VTML (60.7%), and MDM (60.2%)[10, 11, 15, 16]. The improvements to the above scoring matrices are about 3.9%, 6.0%, 4.4%, 4.5%, and 5.0%, respectively. So our method has the highest average coverage for the EUMAT dataset. According to the t-test[17], the coverage of our method is significantly better than the previous results.

2. PRELIMINARIES

2.1. Intrinsically Disordered Proteins

The *intrinsically disordered proteins* (IDPs) are very abundant in nature. The IDPs lack stable tertiary structure under *in vitro* physiological conditions, but the IDPs still have some functions of ordered proteins. The IDPs are involved in many important biological functions, including regulation, recognition, signaling, and control [1]. Brown *et al.* [9] examined the sequence evolution rates of disordered and ordered proteins, and they found that among 26 confirmed protein family abnormalities, 19 families with disordered regions evolved faster than ordered regions, and only 2 families are in opposite.

The intrinsically disordered regions in proteins are significantly different from the amino acid composition of ordered proteins [7]. Uversky [18] calculated the amino acid composition of disordered regions in DisProt, which is a database of intrinsically disordered proteins [19]. Uversky also counted the composition of fully ordered proteins in *Protein Data Bank* (PDB) [20]. He found that the amino acid composition of the disordered regions is different from the ordered regions. In disordered regions, the residues (`A`, `G`, `Q`, `S`, `P`, `E` and `K`) have higher percentage than ordered regions or ordered proteins; the residues (`M`, `H`, `W`, `C`, `F`, `I`, `Y`, `V`, `L` and `N`) have lower percentage in disordered proteins.

The difference in the amino acid composition arises the issue whether the BLOSUM and PAM matrices are suitable for IDPs sequence alignment. Different evolution rates in disordered proteins reflect that it may require different scoring matrices to align IDP sequences.

2.2. The EUMAT Dataset and EDSSMat

The EUMAT dataset are provided by Trivedi and Nagarajaram [3] in 2019. The protein sequences are retrieved from the UniProtKB [21], and the matrices are computed from the disordered alignment blocks at different sequence identity levels. They computed the substitution scoring matrices by implementing the BLOSUM method[10]. Then, the derived matrices are referred to as the matrices of the *dukaryotic disorder substitution scoring matrix* (EDSSMat) series. Table 1 shows the number of proteins and protein superfamilies in the less disordered (LD), moderately disordered (MD) and highly disordered (HD) subsets. Disorder percentage is mean fraction of residues that was predicted to be disordered by the disordered protein predictors [22].

3. THE PROPOSED ALGORITHM

The EDSSMat [3] implements the BLOSUM method [10] on the IDPs. The score of each entry in the widely used scoring matrix ranges from -10 to 20 [23]. Suppose that we want to find the most appropriate symmetric scoring matrix with 23×23, where there are 23 amino acids. In the lower triangle, there are 23 amino acid letters, so there are $\binom{23}{2}$ non-identical letter pairs and 23 identical letter pairs. The total number of entries is $\binom{23}{2} + 23 = 276$. If each entry ranges from -10 to 20, then the number of possible combinations is 31^{276}. Therefore, we try to reconstruct the scoring matrix by using the metaheuristic algorithms. Our goal is to reconstruct a suitable scoring matrix for disordered proteins. It may provide more accurate alignments in homology studies of disordered proteins. In addition, we add the two scores for gap opening and gap extension. Thus, the total number of entries is 278.

In a complex solution space, a good solution may be found quickly by the genetic algorithm (GA). The GA is a global search algorithm based on individual evolution. However, it lacks the effective local search mechanism. When the optimal solution is approaching, the convergence speed is gradually slow or even stagnant. The harmony search (HS) adopts a similar

Table 1: The EUMAT dataset [3], divided into three subsets, including (a) less disordered (LD) (disorder percentage 0% to 20%), (b) moderately disordered (MD) (disorder percentage 20% to 40%), and (c) highly disordered (HD) (disorder percentage >40%).

Dataset	Disorder percentage	Number of proteins	Number of superfamilies
LD	0% to 20%	27832	3352
MD	20% to 40%	5029	1460
HD	>40%	3637	938
Total	–	36498	5750

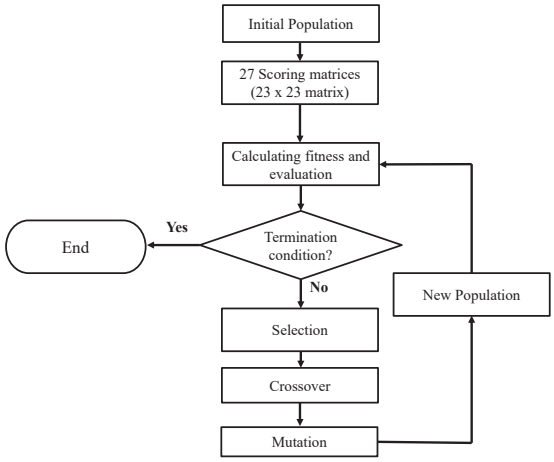


Figure 1: The flow chart of the GAHS algorithm for reconstructing the scoring matrix.

multi-point crossover technique, and it has a fine-tuning mechanism for pitch adjustment to search for neighboring solutions. Therefore, based on the synthesis of the respective advantages of GA and HS [14, 24], we propose the hybrid GAHS algorithm, combining GA and HS, to reconstruct the scoring matrix.

Our GAHS algorithm is an iterative process. First it is initialized with a starting population of chromosomes. A chromosome corresponds to one solution of the scoring matrix. After the initial population has been generated, the GAHS generates for the new population with crossover and mutation and evaluates the fitness with the coverage. It performs several times until it converges to a near best solution. The flow chart of the GAHS algorithm for reconstructing the scoring matrix is shown in Figure 1.

Algorithm: The GAHS algorithm.

Input: k: population size; l: length of a chromosome; R_c: crossover probability; η: mutation probability; G: number of generations.

Output: The scoring matrix with the near highest fitness.

Step 1 (Initialization): The matrix is mapped into a linear array of 276 entries with two additional entries for gap opening and gap extension. From left to right, each entry is assigned by a random value ranging from -10 to 20. The population size k is 27.

Step 2 (Fitness evaluation): Evaluate the fitness of each scoring matrix in the population with the quadratic normalization of coverage measure[25].

Step 3 (Tournament selection): Selection for crossover is performed in order to determine the parents for producing the next generation of chromosomes. The selection is based on the fitness of the chromosome. The more fitness a chromosome, the more likely it will pass parts of its scoring matrices onto the next generation. Selection for crossover is performed by sorting the current population based on its fitness scores. The random number is generated to choose the rank and ultimately pick the chromosome for crossover. As a result, the higher ranked chromosome has a higher probability of crossover, while lower ranked chromosomes has a lower probability of crossover.

Step 4 (Crossover): When a pair of chromosomes is selected for crossover, the two parent matrices will be performed crossover at a randomly selected crossover point to produce two offspring matrices for the next generation.

Step 5 (Mutation): Select a constant number of entries randomly from the new scoring matrix generated by crossover, where the selected number is the $\eta \times 278$. The randomly selected dimension values are given from the harmony memory library in

Equation 1, and then go to pitch adjusting, where the BW is set ±1 in Equation 2.

$$x_i^{\text{new}} = x_i^r, \text{where } x_i^r \in random\{x_i^1, x_i^2, x_i^3, \cdots x_i^{HMS}\}, \quad (1)$$

$$x_i^{\text{new}} = x_i^{\text{new}} \pm U[0,1] \times BW. \quad (2)$$

Step 6 (Termination): If the number of generations reaches the predefined count, then stop; otherwise, go to Step 2.

In each generation, we generate 10 new chromosomes and weed out the poor chromosomes with bad fitness.

4. EXPERIMENTAL RESULTS

Our algorithms are implemented in Python by PyCharm Community Edition 2019.3.2, and the experiments are executed on a computer with Windows 10 64-bit OS, 3.2 GHz CPU (Intel(R) Core(TM) i7-8700 CPU) and 8 GB RAM.

4.1. A Primitive Experiment and Parameter Settings

The parameters of GA, HS and GAHS are shown in Table 2. For a primitive experiment, we build a small dataset of 1000 proteins, extracted from the LD, MD, and HD of the EUMAT dataset with 763, 137, and 100 of proteins, respectively, which preserves the proportion. Furthermore, we test the convergence of the coverage at every setting of mutation probability for 30 times to get the average coverage with the same small dataset of 1000 proteins.

In GA, each case quickly finds a better coverage. But, when η decreases, the chance of getting out of the current solution to find a better solution also decreases. The coverage reaches the higher level until the very late generation. However, the excessively high η may make the algorithm become random search. At about the 12th or 13th generation, the GA enters a stagnation period, and we find that the coverage still rises slowly when $\eta = 0.9$. Thus, GA with $\eta = 0.9$ has the highest coverage. The HS algorithm lacks the crossover operator of GA, which combines two parents. Instead, every decision variable is selected from the existing harmony memory and pitch adjustment is made from it. For the convergence, the GA can converge quickly to a better solution. However, the HS algorithm may jump out of the current solution and find a better coverage. In general, HS with $\eta = 0.1$ has the best performance. The GAHS algorithm uses GA to quickly obtain the better chromosome of the parents and to search for the solution space with mutation. Then the GAHS algorithm refines the bandwidth to search for neighboring solutions with the HS algorithm, instead of simply giving a value randomly. In the primitive experiment, GAHS with $\eta = 0.9$ has the highest coverage. Figure 2 compares the coverages of GA, HS and GAHS for their own best values with $\eta = 0.9$. The GAHS method can obtain more than 80% coverage in the small dataset of 1000 proteins, while the coverages of the other two methods are less than 80%. In summary, the GAHS method, hybrid of GA and HS, improves the performance of GA and HS individually.

Table 2: Parameters of the GA, HS and GAHS algorithms in the primitive experiment.

Algorithm	GA / HS / GAHS
Generation	100
Population size (HM)	27
Decision variables	$278 = \binom{23}{2} + 23 + 2$ (gap opening + gap extension)
Crossover probability (HMCR)	0.9
Mutation probability (PAR)	dynamic adjustment $\eta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$
Termination condition	predefind generation number

We also try to dynamically adjust the value of η. In the initial stage, we use a large mutation probability η to make the initial chromosome be diversified, so that the search solution space range may be wide. In the later stage, in order to converge the overall population, we decrease the mutation probability η. When the generation number is increased, the mutation probability is decreased. Figure 3 shows the coverages of the GA, HS and GAHS with dynamic mutation probability in the primitive experiment. The coverage of GAHS is slightly higher than GA and HS. But, the convergence is very slow after about 10 generations. Compared with dynamic adjustment, setting the mutation probability η to 0.9 can get a slightly higher coverage. However, the experiment for fixed $\eta = 0.9$ requires more time.

4.2. The Experiments of 10-Fold Cross-Validation

The GAHS algorithm builds the scoring matrices for the full EUMAT dataset of $\eta = 0.9$ with dynamic mutation probabilities. We perform the 10-fold cross-validation for 5 times on the full EUMAT dataset. For each 10-fold cross-validation, the EUMAT dataset is

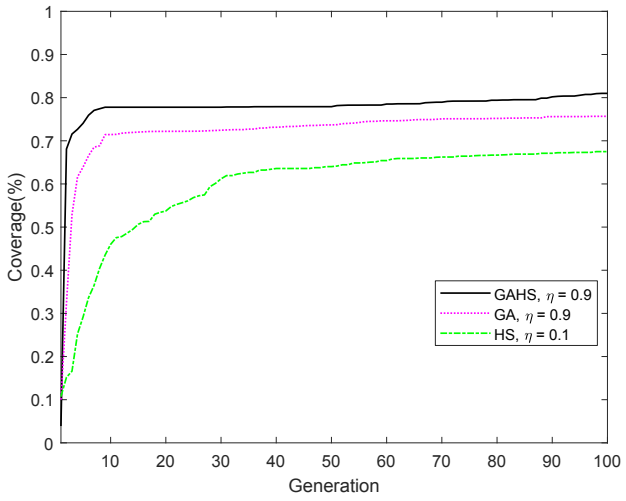


Figure 2: The coverages of the GAHS, GA, and HS algorithms with the best mutation probability η in the primitive experiment.

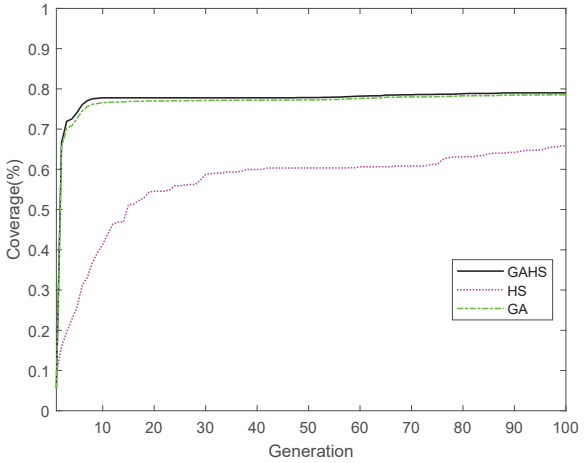


Figure 3: The coverage of the GAHS, GA, and HS algorithms with dynamic mutation probability in the primitive experiment.

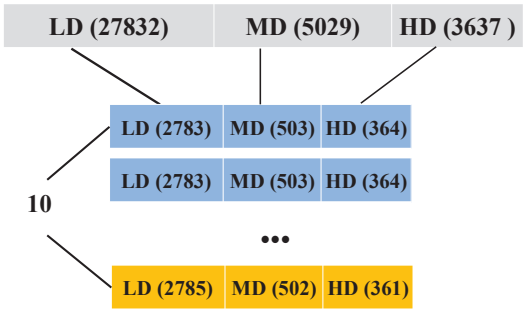


Figure 4: The 10-fold cross-validation construction.

randomly divided into 10 folds, as shown in Figure 4. In one round of each 10-fold cross-validation, it takes 9 folds for training, and the remaining one fold for testing. In the training phase, we apply our GAHS with $\eta = 0.9$ to obtain our scoring matrix. Then we get 50 scoring matrices, where 10 from each 10-fold cross-validation and it is executed 5 times also in our GAHS with dynamic mutation probability. All the training process on the full dataset in 30 generations and the primitive experiment require about 5 days and 2 hours of CPU time, respectively.

Each of the 50 scoring matrices, obtained from our GAHS algorithm of 5 times for 10-fold cross-validation, is performed on the 50 testing folds to obtain the average coverage from 50 folds. With $\eta = 0.9$, the lowest average coverage is 0.5897% (GAHS46), and the highest is 0.6523% (GAHS6). With the dynamic mutation probabilities, the lowest average coverage is 0.6236% GAHS46, and the highest 0.6528% (GAHS77). Then, we obtain the scoring matrix of the highest coverage as our most suitable scoring matrix for the alignment of IDP sequences.

In the testing phase, we take GAHS, EDSSMat, MDM, VTML, BLOSUM, and PAM to obtain the coverage of each testing fold. Thus, 50 coverage percentages are obtained for each scoring matrix. Then the average of the 50 coverages is calculated for comparison. Note that the GAHS scoring matrices are different for different testing folds since it is obtained from the training, while other matrices are always the same for testing folds.

Table 3 shows the average coverages of various scoring matrices. GAHS77 is the best one obtained from our GAHS method. GAHS-average is obtained that each entry is the average of the corresponding entries of the 50 GAHS matrices. The average with "integer" indicates that the value of each entry is rounded to an integer, while "fraction" does not perform the round

Table 3: The average coverages and standard deviations of various scoring matrices with the 50 test folds.

Matrix	Highly disordered	
	Gap open & gap extension	Average coverage
BLOSUM30	−18 & −3	0.5914 ± 0.07
BLOSUM50	−11 & −2	0.5903 ± 0.08
BLOSUM62	**−14 & −3**	**0.5922 ± 0.07**
BLOSUM80	−10 & −3	0.5903 ± 0.08
PAM120	−7 & −1	0.5988 ± 0.08
PAM250	**−19 & −3**	**0.6084 ± 0.07**
MDM10	−18 & −3	0.5513 ± 0.07
MDM20	**−20 & −1**	**0.6027 ± 0.06**
MDM40	−20 & −1	0.5782 ± 0.07
VTML10	−8 & −1	0.5573 ± 0.06
VTML20	−13 & −2	0.5755 ± 0.07
VTML40	−18 & −3	0.5888 ± 0.07
VTML80	−17 & −3	0.6064 ± 0.07
VTML120	**−13 & −3**	**0.6074 ± 0.07**
VTML160	−11 & −2	0.5891 ± 0.08
VTML200	−9 & −3	0.5900 ± 0.08
EDSSMat50	−18 & −2	0.6105 ± 0.07
EDSSMat60	−14 & −3	0.6136 ± 0.08
EDSSMat62	**−19 & −2**	**0.6138 ± 0.07**
EDSSMat70	−19 & −2	0.6131 ± 0.07
EDSSMat75	−19 & −2	0.6135 ± 0.07
EDSSMat80	−15 & −3	0.6110 ± 0.08
EDSSMat90	−19 & −2	0.6120 ± 0.07
GAHS77	**−17 & −2**	**0.6528 ± 0.07**
GAHS-average (integer)	−17 & −2	0.6343 ± 0.08
GAHS-average (fraction)	−17.3 & −2.1	0.6295 ± 0.10

operation, remained as a fraction. Compared with the currently known scoring matrices, our GAHS77 obtains a higher coverage than others. We use t-test to determine whether there is a significant difference between them at a confidence level of 95%, as shown in Table 4. As we can see, the coverage improvements of our matrix compared with other scoring matrices are all statistically significant with t-test.

Figure 5 shows GAHS77 and EDSSMat62. The identical amino acid matches in GAHS are assigned a higher score than EDSSMat62 (such as `RR`, `II`, `LL`), except that the scores of `QQ`, `HH` and `MM` in GAHS are −2, 0, and 7, respectively, while in EDSSMat62, they are 6, 8, and 9, respectively. It is generally recognized that the matching score of identical amino acid should be large, but through the metaheuristic algorithm, it shows that the generated score rules may also be more different. Interestingly, by comparing with the EDSSMat62 matrix, the GAHS77 scoring matrix tends to assign higher

Table 4: The t-test for the coverage difference of our GAHS77 and other scoring matrices obtained from 5 times of 10-fold cross-validation. t-value > 2.0096 and p-value < 0.05 correspond to a 95%-confidence.

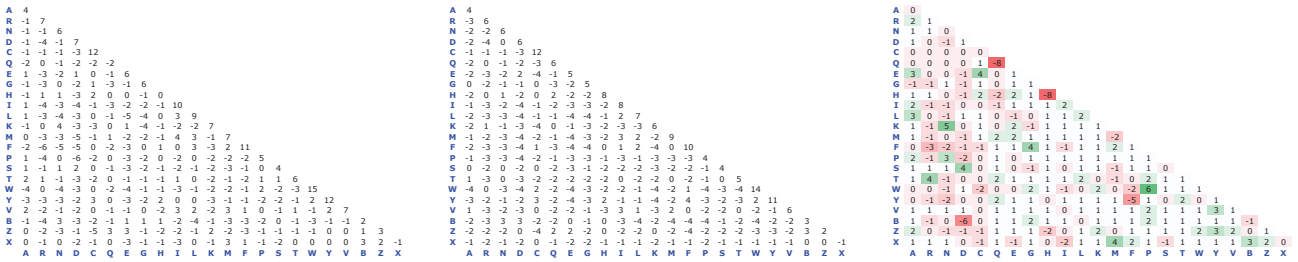
VS	GAHS	
	t-value	p-value (two-tailed)
BLOSUM45	5.078	6.0e-6
BLOSUM50	4.832	1.4e-5
BLOSUM62	4.793	1.6e-5
BLOSUM80	4.748	1.8e-5
PAM120	9.084	4.4e-11
PAM250	6.702	1.9e-8
MDM10	4.224	1.0e-4
MDM20	3.723	5.0e-4
MDM40	4.310	7.8e-5
VTML10	9.840	3.4e-12
VTML20	6.951	7.8e-9
VTML40	5.256	3.0e-6
VTML80	3.855	3.3e-4
VTML120	3.708	5.3e-4
VTML160	4.767	1.7e-5
VTML200	4.850	1.3e-5
EDSSMat50	3.157	2.7e-3
EDSSMat60	3.059	3.5e-3
EDSSMat62	2.908	5.4e-3
EDSSMat70	2.958	4.7e-3
EDSSMat75	2.935	5.0e-3
EDSSMat80	3.115	3.0e-3
EDSSMat90	3.040	3.7e-3

scores (`KN`, `WP`, `SD`, `TR`, etc.) for different matching, which is more likely to happen due to the different amino acid composition of disordered and ordered proteins.

In Table 5, we can see that average differences of disorder-promoting residues (`A`, `G`, `Q`, `S`, `P`, `E` and `K`) is higher than BLOSUM62 and PAM250. On the contrary, the average differences of order-promoting residues (`M`, `H`, `W`, `C`, `F`, `I`, `Y`, `V`, `L` and `N`) is higher than EDSSMat62. Thus, we confirm that the scoring matrix may have different scores with respect to the composition of disordered proteins.

5. CONCLUSION

In the past decades, several reconstruction methods have been proposed for building various scoring matrices for protein sequence alignment. These scoring matrices are mainly for the general proteins, For IDP proteins, the EDSSMat has been constructed[3], and it still adopts the BLOSUM method[10]. This paper reconstructs the scoring matrix by using metaheuristic algorithms to improve IDP homology searching. The



(a) GAHS77. (b) EDSSMat62. (c) GAHS77 minus EDSSMat62.

Figure 5: The comparison of scoring matrices of GAHS77 and EDSSMat62.

experimental results on the EUMAT dataset show that the average coverage of our reconstructed scoring matrix improves about 3.9%, 6.0%, 4.4%, 4.5%, and 5.0% compared to EDSSMat, BLOSUM, PAM, VTML, and MDM, respectively, and it is statistically significant with t-test.

According to the experimental results, it is possible to improve. In our experiments, most of the overall execution time is spent on the sequence alignment, which is the key step for calculating the fitness of a scoring matrix to detect homology. It may be worthy to develop more efficient sequence alignment algorithms. In addition, we can try other metaheuristic algorithms, such as *search economics*, to modify the scores on some regions, instead of the entire scoring matrix.

REFERENCES

[1] 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.

[2] P. E. Wright and H. J. Dyson, "Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm," *Journal of Molecular Biology*, vol. 293, no. 2, pp. 321–331, 1999.

[3] 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.

[4] C. J. Brown, A. K. Johnson, and G. W. Daughdrill, "Comparing models of evolution for ordered and disordered proteins," *Molecular Biology and Evolution*, vol. 27, no. 3, pp. 609–621, 2010.

[5] U. Midic, A. K. Dunker, and Z. Obradovic, "Protein sequence alignment and structural disorder: a substitution matrix for an extended alphabet," in *Proceedings of the KDD-09 Workshop on Statistical and Relational Learning in Bioinformatics*, 2009, pp. 27–31, new York, United States.

[6] P. Radivojac, Z. Obradovic, C. J. Brown, and A. K. Dunker, "Improving sequence alignments for intrinsically disordered proteins," in *Biocomputing 2002*. World Scientific, 2001, pp. 589–600.

[7] P. Romero, Z. Obradovic, X. Li, E. C. Garner, C. J. Brown, and A. K. Dunker, "Sequence complexity of disordered protein," *Proteins: Structure, Function, and Bioinformatics*, vol. 42, no. 1, pp. 38–48, 2001.

[8] A. M. Szalkowski and M. Anisimova, "Markov models of amino acid substitution to study proteins with intrinsically disordered regions," *PLoS one*, vol. 6, no. 5, 2011.

[9] C. J. Brown, S. Takayama, A. M. Campen, P. Vise, T. W. Marshall, C. J. Oldfield, C. J. Williams, and A. K. Dunker, "Evolutionary rate heterogeneity in proteins with long disordered regions," *Journal of Molecular Evolution*, vol. 55, no. 1, p. 104, 2002.

[10] 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. 10 915–10 919, 1992.

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

[12] V. Vacic, V. N. Uversky, A. K. Dunker, and S. Lonardi, "Composition profiler: a tool for discovery and visualization of amino acid composition differences," *BMC Bioinformatics*, vol. 8, no. 1, p. 211, 2007.

[13] Holland and J. Henry, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, 1992.

[14] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[15] 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.

[16] 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.

Table 5: The average differences of each residue pairs between two scoring matrices and the average scores of EDSSMat62 and GAHS77.

Amino acid symbols	Average differences			Average scores	
	GAHS77 VS BLOSUM62	GAHS77 VS PAM250	GAHS77 VS EDSSMat62	EDSSMat62	GAHS77
A	0.6 ± 1.1	0.4 ± 1.3	1.0 ± 1.0	−1.3 ± 1.7	−0.3 ± 1.8
G	0.6 ± 1.6	0.3 ± 2.4	0.7 ± 1.2	−1.8 ± 1.9	−1.1 ± 2.0
Q	−0.4 ± 1.8	−0.3 ± 2.5	0.1 ± 2.0	−1.0 ± 2.0	−0.9 ± 1.4
S	−0.2± 1.3	−0.3 ± 1.5	0.7 ± 1.0	−1.3 ± 1.5	−0.7 ± 1.7
P	0.7 ± 2.2	0.1 ± 2.6	1.2 ± 1.5	−2.1 ± 1.7	−1.0 ± 2.2
E	−0.5 ± 1.7	−0.3 ± 2.1	0.8 ± 1.2	−1.9 ± 2.3	−1.0 ±2.5
K	−0.2± 1.4	−0.1± 1.7	1.0 ± 1.2	−1.9 ± 2.2	−0.9 ± 2.5
H	0.4 ± 2.4	−0.2 ± 2.4	0.2 ± 2.1	−0.6 ± 2	−0.3 ± 1
M	0.4 ± 1.6	0.5 ± 1.7	0.7 ± 1.3	−1.0 ± 3	−0.4 ± 3
C	1.4 ± 1.9	2.7 ± 2.1	0.3 ± 1.2	−0.6 ± 3.3	−0.3 ± 3.2
W	1.0 ± 1.8	2.5 ± 2.7	0.6 ± 1.6	−1.4 ± 3.9	−0.8 ± 3.8
F	0.5 ± 2.1	1.1 ± 2.7	0.2 ± 1.9	−1.0 ± 3.3	−0.9 ± 3.4
I	0.9 ± 1.6	0.3 ± 1.6	0.3 ± 1.1	−0.9 ± 2.7	−0.6 ± 3.2
Y	0.7 ± 2.0	1.4 ± 2.4	0.5 ± 1.6	−0.8 ± 3.3	−0.3 ± 3.2
V	1.5 ± 0.8	1.2 ± 1.4	1.1 ± 0.6	−0.8 ± 2.2	0.3 ± 2.1
N	−0.1 ± 1.6	−0.4 ± 1.9	0.2 ± 1.6	−1.0 ± 2.1	−0.3 ± 3.1
L	1.1 ± 1.5	1.1 ± 1.7	0.8 ± 0.9	−1.1 ± 2.6	−0.8 ± 2.6
R	−0.6 ± 1.8	−1.0 ± 1.8	0.1 ± 1.4	−1.7 ± 2.1	−1.6 ± 2.6
D	−1.1 ± 2.1	−1.3 ± 2.7	−0.3 ± 1.7	−1.8 ± 3	−2.1 ± 3
T	0.4 ± 1.1	0.3 ± 1.4	1.0 ± 1.1	−1.3 ± 2	−0.3 ± 2
B	0.0 ± 2.1	−0.3 ± 2.1	0.5 ± 1.7	−1.4 ± 2	−0.9 ± 2
Z	0.0 ±1.6	0.1 ± 2.4	0.7 ± 1.2	−1.2 ± 2	−0.5 ± 2
X	0.7 ± 1.6	0.8 ± 1.9	0.8 ± 1.3	−1.1 ± 1	−0.3 ± 2

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

[18] V. N. Uversky, "The alphabet of intrinsic disorder: II. various roles of glutamic acid in ordered and intrinsically disordered proteins," *Intrinsically Disordered Proteins*, vol. 1, no. 1, pp. 1–23, 2013.

[19] M. Sickmeier, J. A. Hamilton, T. LeGall, V. Vacic, M. S. Cortese, A. Tantos, B. Szabo, P. Tompa, J. Chen, and V. N. Uversky, "Disprot: the database of disordered proteins," *Nucleic Acids Research*, vol. 35, no. suppl_1, pp. D786–D793, 2007.

[20] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[21] 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," in *Plant Bioinformatics*. Springer, 2016, pp. 23–54.

[22] P. Lieutaud, F. Ferron, A. V. Uversky, L. Kurgan, V. N. Uversky, and S. Longhi, "How disordered is my protein and what is its disorder for? a guide through the "dark side" of the protein universe," *Intrinsically Disordered Proteins*, vol. 4, no. 1, pp. 275–282, 2016.

[23] W. R. Pearson, "Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms," *Genomics*, vol. 11, no. 3, pp. 635–650, 1991.

[24] A. Thengade and R. Dondal, "Genetic algorithm-survey paper," in *MPGI National Multi Conference*. Citeseer, 2012, pp. 7–8.

[25] R. E. Green and S. E. Brenner, "Bootstrapping and normalization for enhanced evaluations of pairwise sequence comparison," *Proceedings of the IEEE*, vol. 90, no. 12, pp. 1834–1847, 2002.