Flexible Protein Structure Alignment Based on Topology String Alignment of Secondary Structure

Jafar Razmara, Samira Fotoohi, and Sepideh Parvizpour

***Abstract*—The paper reports a method for flexible alignment of protein structure. The method in the first phase applies a text modeling technique to obtain an initial superposition of secondary structure elements of two proteins. Then, in the second phase, a step-by-step algorithm is utilized to create flexible alignment between two structures. The method was assessed using a dataset of proteins with macromolecular motions and the results compared with those of the existing flexible alignment methods e.g. FlexProt, FATCAT, and FlexSnap. The results demonstrate that the method have a competitive accuracy in comparison with the other similar methods.**

***Index Terms*—Protein structure alignment, flexible alignment, structure comparison.**

I. Introduction

Structural similarity analysis of proteins has been a major classical challenge in structural biology for more than two past decades. It has wide applications in classification, functional annotation and evolutionary relationships analysis. Many studies have been done during past decades to compare and align protein structures as rigid objects. However, biologists believe that proteins have a flexible structure and go through conformational changes in order to do their normal functions [1], [2]. Therefore, in order to have a significant structural comparison, it is necessary to consider flexibility for the molecules having undergone conformational change.

The majority of existing methods for protein structure comparison supposes these biomolecules as rigid body [3]-[8], and the problem of flexible protein structure comparison has not been received enough attention. The problem is formulated as to find the optimal structural alignment between two proteins with the least number of rearrangement or twist in one of the structures [9]. Several algorithms have been developed to solve the problem. FATCAT [9] as a known flexible alignment method works based on clustering, where it firstly produces aligned fragment pairs (AFPs), and then, allows flexibility while making a chain of AFPs using dynamic programming. FlexProt [10] is another scheme that looks for the longest chain of AFPs having different number of hinges. HingeProt [11] firstly divides one of the proteins into rigid parts using an approach based on Gaussian-Network-Model, and then, applies MultiProt [12] to align each part with the second protein. RAPIDO [13] uses a flexible aligner that is coupled to a genetic algorithm for the identification of structurally conserved regions. It is capable of aligning protein structures in the presence of large conformational changes. Structurally conserved regions are reliably detected by RAPIDO even if they are discontinuous in sequence but continuous in space and can be used for superpositions revealing subtle differences. Moreover, FlexSnap [14] is a greedy chaining algorithm for flexible sequential and non-sequential alignment of protein structure. The main idea used in the FlexSnap algorithm is to assemble short well-aligned AFPs. FlexSnap has shown a competitive effectiveness in the assessments in comparison with the other state of the art flexible alignment methods by considering non-sequential alignment of the structures.

In the recent years, a number of methods have been developed based on linear encoding of protein structure [15]-[18]. The methods generally encode protein structure into linear sequences, and then, apply sequence alignment techniques to align two structures. The main idea in development of these methods is to speed up the homology search within a database of protein structures. However, the methods obtain lower accuracy than state of the art geometry based methods. We have developed recently a topology string based method [18] which uses both linear encoding and geometry based schemes to align two protein structures. Thus, the method obtains high running speed as well as linear encoding based methods, while it has a competitive accuracy with geometry based algorithms. Based on the fruitful results of the method, now we extend the scheme for flexible alignment of protein structure.

II. Methods

The proposed method in this paper works in two main phases. In the first phase, the method superposes secondary structure elements of two structures to achieve an initial overlap between two structures. Moreover, in the second phase, the method uses a step-by-step algorithm to align two structures considering flexibility of the AFPs. The following is the detailed description of the method.

A. Secondary Structure Superposition

Secondary structure of a protein is known as the backbone of a protein and is made of highly regular substructures called α-helix and β-strand. To achieve an initial overlap between two structures, the method encodes geometry of secondary

Manuscript received June 15, 2013; revised August 7, 2013. We would like to thank our sponsor the Malaysian Ministry of Science, Technology, and Innovation (MOSTI) for supporting this research under GUP grant Q.J130000.2528.04H91.

J. Razmara and S. Fotoohi are with the Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia (e-mail: jafar@utm.my, samira.fotoohi@gmail.com).

S. Parvizpour is with the Faculty of Biosciences and Medical Engineering, Universiti Teknologi Malaysia, Johor, Malaysia (e-mail: se.parvizpour@gmail.com).

structure elements (SSEs) of a protein in a topology string called SSEs sequence according to the scheme introduced by the authors in [18]. To this end, each element is assumed as a vector $r_{SSE}=r_b-r_e$ where

$$r_b = (0.74r_i + r_{i+1} + r_{i+2} + 0.74r_i+3) / 3.48,$$
$$r_e = (0.74r_{j-3} + r_{j-2} + r_{j-1} + 0.74r_j) / 3.48 \quad (1)$$

for helices and

$$rb = (r_i + r_{i+1}) / 2,\ r_e = (r_{j-1} + r_j) / 2 \quad (2)$$

for strands [8] (indices *i* and *j* denote the first and last residues in each element). Based on the sign of the *x*, *y*, and *z* components, each vector is encoded to a letter as shown in Table I. Moreover, for each pair of consecutive SSE vectors, an inter-SSE vector is defined using end and start points of two SSEs. This vector determines relative position of an element with respect to its previous vector. Fig. 1 shows a typical example for SSEs representation as a set of vectors and their encoding in a topology string.

In order to find a correspondence between two structures, it is necessary to rotate a structure around the other or use a coordinate independent representation of two structures. The method applies a string permutation scheme according to the technique introduced in [18]. The scheme generates 24 permuted strings from topology string of each protein by rotating the structure 90 degree around the *x*, *y*, and *z* axes. For each rotation around axes, the letters in the topology string are permuted according to Table II.

The above 24 permuted strings are considered as estimation of different possible orientation of a query structure that can be matched with the reference proteins within a database of protein structure. To find a match between query structure and each reference protein structure, the method applies cross entropy measure over n-gram modeling technique derived from computational linguistics as a superior technique to any formal language modeling approaches [19]. The technique firstly makes n-gram model by counting the words of one sequence in the training phase, and then, measures predictability of the second sequence in the recall phase via formula:

TABLE I: SECONDARY STRUCTURE VECTORS DIRECTION AND THEIR DEFINED LABELS

Direction			Vector type		
			Strand	**Helix**	**Inter-SSEs**
+*x*	+*y*	+*z*	A	I	Q
+*x*	+*y*	-*z*	B	J	R
+*x*	-*y*	+*z*	C	K	S
+*x*	-*y*	-*z*	D	L	T
-*x*	+*y*	+*z*	E	M	U
-*x*	+*y*	-*z*	F	N	V
-*x*	-*y*	+*z*	G	O	W
-*x*	-*y*	-*z*	H	P	X

TABLE II: PERMUTATION OF THE LETTERS BASED ON 90 DEGREE ROTATION AROUND *X*, *Y*, AND *Z* AXES

	Strand	**Helix**	**Inter-SSEs**
Old	ABCDEFGH	IJKLMNOP	QRSTUVWX
Rotate around *x*	BDACFHEG	JLIKNPMO	RTQSVXUW
Rotate around *y*	EAGCFBHD	MIOKNJPL	UQWSVRXT
Rotate around *z*	EFABGHCD	MNIJOPKL	UVQRWXST

$$H(X, P_M) = -\sum_{\text{all } w_i^n} P(w_i^n) \log(2 + P_M(w_{i+n}|w_i^{n-1}))$$
$$= -\frac{1}{N}\sum_{\text{all } w_i^n} \text{Count}(w_i^n) \log(2 + P_M(w_{i+n}|w_i^{n-1})) \quad (3)$$

where the variable *X* is in the n-gram form $w_i^n=\{w_i, w_{i+1}, \ldots, w_{i+n-1}\}$. The summation runs over all the possible *n*-gram words w_i^n, and *N* is the number of *n*-grams. The term $P(w_i^n)$ is computed by the word count within the first sequence via $\text{Count}(w_i^n)$. Moreover, the conditional probability in the summation makes relation between the *n-th* element of an n-gram and the preceding *n*-1 elements, which can be computed by counting the words of the second sequence and having the model estimated:

$$P(w_{i+n}|w_i^{n-1}) = \text{Count}(w_{i+n})/\text{Count}(w_i^{n-1}) \quad (4)$$

The above cross entropy formula is used to measure similarity of each 24 different topology strings of the query structure with the topology string of each reference protein via the formula:

$$D(S_r, S_i) = |H(X_r, P_{M_i}) - PS| \quad (5)$$

where *PS* is the perfect score using the first sequence as reference and model sequences. S_r and S_i also denote the reference topology string and *i*-th string of query structure respectively. The lower value of D (S_r, S_i) indicates higher similarity of the compared sequences.

Having the most similar permuted topology string of a query structure to that of a reference protein, now, a procedure matches identical n-gram words of the two strings. The procedure applies an iterative task for decreasing size of n-grams from m (chosen empirically 6) down to basic size of n-grams (chosen at 3). After that, the procedure makes another effort to match letters belong to the SSEs with semi-adjacent vectors, where they have only one component with different sign. As the result, the procedure creates the map of correspondence between SSEs of two compared proteins.

B. Flexible Structure Alignment

Having the map of correspondence between SSEs of two structures, in the second phase, the method aligns two proteins at the residue level. A step-by-step procedure is used by the method to find the optimal flexible alignment by assembling AFPs and introducing hinges. The main goal in this procedure is to find a maximal size of alignment and minimal number of flexible regions between two structures. The procedure firstly proceeds to align two structures as rigid body in the steps 1-3, and then, tries to find flexible regions between two structures to introduce hinges. Moreover, it uses the rotation matrix of Kabsch method [20] to obtain optimal correspondence between aligned pairs of residues.

1) For each matched SSE pair, choose start and end residues as temporarily aligned residues. Compute and apply the Kabsch rotation matrix to achieve an initial correspondence between two structures.
2) Clear the list. For each matched SSE pair, choose *n* neighboring residues (*n* is chosen 3 for strands and 4 for helices [6]) having minimum distance within each matched SSE pair. Extend the alignment for residues

with a distance less than a certain threshold *t*.

3) Make Kabsch rotation matrix based on the alignment list and apply to whole structure of the first protein.
4) Extend the alignment for unaligned residues between the aligned pairs by finding contact pairs [6] having the maximum distance *t*. Mark the aligned parts as AFP.
5) Apply a dynamic programming algorithm to find the optimal chaining of AFPs obtained in step 4: Add AFP *k*+1 to *k* previous chained AFPs by scoring long AFPs and penalizing large RMSDs and gaps between two connected AFPs. Introduce a twist to add an AFP *k*+1 to the chain if it is not compatible [9] with AFP *k*.

After the above alignment task, the alignment result is processed by a refinement procedure. The procedure introduces more twists into the chain if the overall RMSD is less than a defined threshold. Moreover, twists that increase the overall RMSD more than a threshold are removed.

III. Results

The above algorithm has been implemented in C++ and run on a 2.40 GHz Intel Core i3 with 4 GB of main memory running Microsoft Windows 7. The running time for comparing each pair varies from 0.1 second to 10 seconds depending on protein size and number of AFPs of two structures.

To test quality of the presented method, we first applied the method to align a set of protein pairs that is described as 'difficult' alignment in the literature [21]. The results are compared with three rigid alignment programs including DALI [3], VAST [22], and CE [5] and represented in Table III. Based on the results, our presented method generally gives higher length of alignment in comparison with three other methods. Actually, higher length of alignment cause an increase in the RMSD value computed for the alignment. However, the method has a competitive accuracy in terms of RMSD with three others.

The quality of the introduced method was assessed in comparison with other flexible structure alignment methods including FlexProt, FATCAT, and FlexSnap. The assessment compares the outputs of the methods on the FlexProt dataset [10] which is collected from the dataset of macromolecular motions [23]. The results of the assessment are shown in Table IV where the data for three other methods were taken from [14]. In this table, the parameter *T* is the number of hinges in the alignment. From the results, it is observed that the method generates competitive results with FlexSnap and FATCAT methods. In some cases, the method gives lower length of alignment with lower RMSD value. In general, the results of assessment prove that the method works well in comparison with three other flexible alignment methods.

IV. Conclusion

A flexible protein structure alignment method was introduced in this paper. The method uses topology string alignment of secondary structure elements to achieve an initial overlap between two structures. In the second phase, a step-by-step algorithm is used to create the alignment. Based on the assessment on a dataset of proteins with macromolecular motions, it is demonstrated that the method has high efficiency in comparison with the other similar methods.

TABLE III: Comparing Structure Alignment Results of 10 'Difficult' Pairs of Structures from Fischer Dataset by Different Methods

		VAST		DALI		CE		Our method	
Protein 1	Protein 2	Length	RMSD	Length	RMSD	Length	RMSD	Length	RMSD
1fxiA	1ubq	48	2.1	-	-	-	-	54	2.41
1ten	3hhrB	78	1.6	86	1.9	87	1.9	87	1.90
3hlaB	2rhe	-	-	63	2.5	85	3.5	82	3.21
2azaA	1paz	74	2.2	-	-	85	2.9	85	2.91
1cewI	1molA	71	1.9	81	2.3	69	1.9	79	2.20
1cid	2rhe	85	2.2	95	3.3	94	2.7	98	2.97
1crl	1ede	-	-	211	3.4	187	3.2	245	3.14
2sim	1nsbA	284	3.8	286	3.8	264	3.0	276	3.18
1bgeB	2gmfA	74	2.5	98	3.5	94	4.1	97	3.23
1tie	4fgf	82	1.7	108	2.0	116	2.9	112	2.96

TABLE IV: Comparing Alignment Results of Our Method with the Results of FlexProt, FATCAT, and FlexSnap Methods

		FlexProt			**FATCAT**			**FlexSnap**			**Our method**		
Protein 1	**Protein 2**	Length	RMSD	*T*	Length	RMSD	*T*	Length	RMSD	*T*	Length	RMSD	*T*
1wdnA (223)	1gggA (220)	218	0.94	2	220	1.01	2	220	0.96	2	220	0.98	2
1hpbP (238)	1gggA (220)	220	2.34	2	213	1.59	2	211	1.67	2	211	1.71	2
2bbmA (148)	1cll_ (144)	139	2.22	1	144	2.28	1	138	1.8	1	135	1.74	1
2bbmA (148)	1top_ (162)	147	2.40	3	145	2.28	3	137	1.78	3	132	1.81	3
1akeA (214)	2ak3A (226)	200	2.44	2	202	1.54	2	207	2.05	2	205	2.1	2
2ak3A (226)	1uke_ (193)	182	2.90	2	188	2.97	0	184	2.36	1	185	2.44	1
1mcpL (220)	4fabL (219)	218	1.93	1	217	1.40	1	217	1.49	1	218	1.42	1
1mcpL (220)	1tcrB (237)	212	2.33	1	213	2.20	1	202	2.3	1	206	2.29	1
1lfh (691)	1lfg_ (691)	691	1.41	2	686	0.89	2	688	0.99	2	684	092	2
1tfd (294)	1lfh_ (691)	291	1.98	2	290	1.37	2	287	1.89	2	290	1.81	2
1b9wA (91)	1danL (142)	75	2.78	1	80	2.39	2	82	2.25	2	80	2.31	2
1qf6A (641)	1adjA (420)	323	4.43	1	351	2.68	1	326	2.45	3	321	2.39	1
2clrA (275)	3fruA (269)	253	2.71	2	245	3.06	0	254	2.57	3	248	2.62	2
1fmk (438)	1qcfA (450)	424	1.25	2	433	2.27	0	413	2.71	0	423	2.93	0
1fmk (438)	1tkiA (321)	231	3.28	2	238	3.07	0	241	2.58	3	238	2.89	2
1a21A (194)	1hwgC (191)	163	2.75	4	153	3.16	1	156	2.35	3	152	2.76	3

ACKNOWLEDGMENT

We would like to thank our research grant sponsors, Malaysian Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM) for their support (research vote number: 04H91).

REFERENCES

[1] W. Bennett and R. Huber, "Structural and functional aspects of domain motions in proteins," *Crit. Rev. Biochem.*, vol. 15, pp. 291–384, 1984.
[2] D. J. Jacobs, A. J. Rader, L. A. Kuhn, and M. F. Thorpe, "Protein flexibility predictions using graph theory," *Proteins*, vol. 44, pp. 150–165, 2001.
[3] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," *Journal of Molecular Biology*, vol. 233, pp. 123-138, 1993.
[4] J. F. Gibrat, T. Madej, J. L. Spouge, and S. H. Bryant, "The VAST protein structure comparison method," *Biophysics Journal*, vol. 72, MP298, 1997.
[5] I. Shindyalov and P. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path," *Protein Engineering*, vol. 11, pp. 739-47, 1998.
[6] E. Krissinel and K. Henrick, "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions," *Acta Crystallographica Section D: Biological Crystallography*, vol. 60, pp. 2256-2268, 2004.
[7] Y. Zhang and J. Skolnick, "TM-align: a protein structure alignment algorithm based on the TM-score," *Nucleic Acid Research*, vol. 33, pp. 2302-2309, 2005.
[8] B. Kolbeck, P. May, T. Schmidt-Goenner, T. Steinke, and E. W. Knapp, "Connectivity independent protein-structure alignment: A hierarchical approach," *BMC Bioinformatics*, vol. 7, doi:10.1186/1471-2105-7-510, 2006.
[9] Y. Ye and A. Godzik, "Flexible structure alignment by chaining aligned fragment pairs allowing twists," *Bioinformatics*, vol. 19, pp. II246-II255, 2003.
[10] M. Shatsky, R. Nussinov, and H. Wolfson, "Flexible protein alignment and hinge detection," *Proteins: Structure, Function, and Bioinformatics*, vol. 48, pp. 242-256, 2002.
[11] U. Emekli, D, Schneidman-Duhovny, H. Wolfson, R. Nussinov, and T. Haliloglu, "HingeProt: Automated prediction of hinges in protein structures," *Proteins*, vol. 70, pp. 1219-1227, 2008.
[12] M. Shatsky, R. Nussinov, and H. Wolfson, "A method for simultaneous alignment of multiple protein structures," P*roteins: Structure, Function,and Bioinformatics*, vol. 56, pp. 143-156, 2004.
[13] R. Mosca and T. Schneider, "RAPIDO: A web server for the alignment of protein structures in the presence of conformational changes," *Nucleic Acids Research*, vol. 36, W42-W46, 2008.
[14] S. Salem, M. J. Zaki, and C. Bystroff, "FlexSnap: Flexible non-sequential protein structure alignment," *Algorithms for Molecular Biology*, vol. 5, 2010.
[15] M. Carpentier, S. Brouillet, and J. Pothier, "YAKUSA: A fast structural database scanning method," *Proteins*, vol. 61, pp. 137-151, 2005.
[16] C. H. Tung, J. W. Huang, and J. M. Yang, "Kappa-alpha plot derived structural alphabet and BLOSUM-like substitution matrix for rapid search of protein structure database," *Genome Biology*, vol. 8:R31, 2007.
[17] W. C. Lo, P. J. Huang, C. H. Chang, and P. C. Lyu, "Protein structural similarity search by Ramachandran codes," *BMC Bioinformatics*, vol. 8:307, 2007.
[18] J. Razmara, S. Deris, and S. Parvizpour, "TS-AMIR: A topology string alignment method for intensive rapid protein structure comparison," *Algorithms for Molecular Biology*, vol. 7, 2012.
[19] A. Bogan-Marta, A. Hategan, and I. Pitas, "Language engineering and information theoretic methods in protein sequence similarity studies," *Studies in Computational Intelligence*, vol. 85, pp. 151-183, 2008.
[20] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 34, pp. 827-828, 1978.
[21] D. Fischer, A. Elofsson, D. Rice, and D. Eisenberg, "Assessing the performance of fold recognition methods by means of a comprehensive benchmark," *Pacific Symposium on Biocomputing*, pp. 300–318, 1996.
[22] T. Madej, J. F. Gibrat, and S. H. Bryant, "Threading a database of protein cores," *Proteins*, vol. 23, pp. 356–369, 1995.
[23] M. Gerstein and W. Krebs, "A database of macromolecular motions," *Nucleic Acids Res*, vol. 26, pp. 4280-4290, 1998.

Jafar Razmara received his B.Sc. in 1995 from Isfahan University of Technology, M.Sc. in 1998 from Tarbiat Modares University both in Software Engineering, and PhD. in 2012 from Software Engineering department of Faculty of Computing at University Technology Malaysia. He is presently a senior lecturer in the Faculty of Computing at University Technology Malaysia. His main research interests are bioinformatics, computational biology, artificial intelligence, data mining and soft computing.

Samira Fotoohi received her B.Sc. in 2004 from Najafabad IAU University in Computer engineering and M.Sc. in 2012 from University Technology Malaysia in Information Technology. She is now a PhD candidate in the Faculty of Computing. Her main research interests are bioinformatics, computational biology, artificial intelligence, data mining.

Sepideh Parvizpour received her B.Sc. in 1999 from Tabriz University in Biology, and M.Sc. in 2011 from University Technology Malaysia in Biotechnology. She is now a PhD candidate in the Faculty of Biosciences and Medical Engineering. Her main research interests are bioinformatics, computational biology and biotechnology.