Intelligent Protein 3D Structure Retrieval System

Yiqiang Chen[1] ,Wen Gao[1,2], Lijuan Duan[1], Xiang Chen[1],Charles X. Ling[3]

[1](Institute of Computing Technology, Chinese Academy of Sciences, 100080)
[2](Department of Computer Science, Harbin Institute of Technology, 150001)
[3](Department of Computer Science, University of West Ontario, CA)
(yqchen@ict.ac.cn;86-10-82649008;86-10-82649298)

Abstract. Since the 3D structure of a protein determines its function, the protein structural identification and comparison system is very important to biologists. In this paper, an intelligent protein 3D structure retrieval system is described. The system is intelligent since it integrates the moment feature extraction technology and the relevant feedback method in Artificial Intelligence (AI). As there is no universal agreement on the similarity of proteins structures, the major advantage of our system compared to other previous systems is that we use the relevance feedback technology to aid the biologists to find the similar protein structures more effectively. The similarity metric formula is improved dynamically by biologists' interaction through relevance feedback. The experimental results show that the proposed approach can capture the biologists' intentions in real-time and obtain good performance in the protein 3D structure retrieval. The ratio of total improvement is about 15.5% on average, which is quite significant compared to the improvements obtained in some previous work.

1. INTRODUCTION

Most biological actions of proteins, such as catalysis or regulation of the genetic messages, depend on certain particular components of their three-dimension (3D) structures. Proteins with similar 3D structures often show similar biological properties, or have the same functions [1]. It is therefore highly desirable to measure the similarities between protein 3D structures. One of the primary goals of protein structural alignment programs is to quantitatively measure the level of structural similarity between pairs of known protein structures.

There have been several previous methods that compare protein structures and measure the degree of structural similarity between them. For instance, 3dSEARCH [2] is based on geometric hashing, an object recognition algorithm developed in the field of computer vision. Dali Server [3] presents a general approach to aligning a pair of proteins represented by two-dimension distance matrixes. Another well-known algorithm called Combinatorial Extension (CE) defines aligned fragment pairs (AFPs) to confer structure similarities of proteins [4]. In recent years, the method of moments and mesh representation for 3D model retrieval, which succeeds in computer vision [5], is adapted and extended to perform retrieval of 3D protein structures [6].

Since there is no universal agreement on the similarity of proteins, it is not easy to assess the results of similarity retrieval systems to tell which one is the best [7]. The biologists intend to use their own similarity measure to formulate their queries. In the systems mentioned above, the parameters and methods of feature extraction and similarity measurement are pre-determined, and they cannot be adjusted intelligently according to the experimental conditions. Thus the biologists' subjective perceptions could not effectively be modeled by the features and their associated weights used in assessing protein similarities. To resolve this problem, relevance feedback is adapted in our system. Relevance feedback is a powerful tool, and it has been successfully used in text or image retrieval [8,9,10,11]. Retrieval system based on the relevance feedback allows users to provide coarse queries initially. The system provides an interface to allow the users to decide which answers are correct or incorrect, and then the system learns from the positive (correct) and negative (incorrect) examples submitted by the users, and submit a more accurate set of answers to the users. The process may iterate several times until users are satisfied with the answer. As it is often difficult for users to express their intention, the approach of relevance feedback will be very effective in retrieving proteins that users have desired.

A new intelligent protein 3D structure retrieval system, integrating moment feature extracting technology and relevance feedback method, is proposed in this paper. Figure 1 shows the framework of our intelligent protein 3D structure retrieval system. It not only obtains 3D features but also refines the queries and the similarity metric in the retrieval process. The similarity metric formula can be improved dynamically according to users' interaction with the system. Experiments show that this system can satisfy users' queries.

Two key technologies used in our system can be further explained here. One is the feature extraction technology (see Section 2). From the PDB [13], we can extract two kinds of features: protein 3D structural features (such as the geometric shape feature or the moment [6]) and protein description features (such as the number of atoms, the weight of each atoms, and the name of each atoms). The features can be normalized for similarity measurement.

The other key technology is relevance feedback (see Section 3), which, as far as we know, has not been applied in protein retrieval systems previously. Relevance feedback allows users to iteratively retrieve similar protein structures from any large protein 3D structure database, such as PDB. A user can browse the protein database and investigate the detailed structure for each protein. Once he/she finds a protein structure of interests, that protein structure is submitted as a query. The system computes the similarity between proteins in the database and the query protein, and ranks them according to the similarity scores. If the user wants to improve the retrieval results, he can submit some positive or negative examples to the system. The system computes the similarity scores according to our relevance feedback method [12], and ranks proteins again. The above process is repeated until user is satisfied with the results.

Figure 2 illustrates a concrete query process with relevance feedback. In Figure 2, the query protein is displayed at the upper left corner, and the best 20 retrieved proteins are displayed in the right window (but only 9 of them are shown here due to picture size). Users can put a check mark or cross mark on each protein retrieved, and then the proteins with the check marks on them are submitted as positive examples,

while the proteins with the cross marks are submitted as negative examples. The system takes in the users' feedback, adjusts the feature weights and the similarity metric formula (see Section 3), and re-computes the similarity score and produces a new ranked list that will contain more proteins with the similar features as the user has chosen. This process may repeat several times until the user is satisfied.

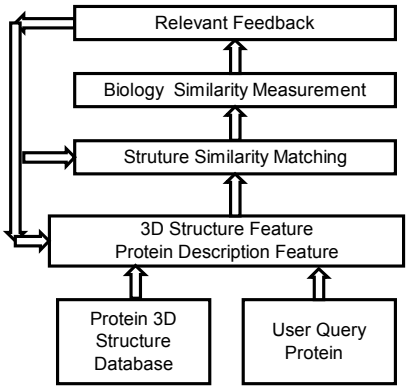

Fig. 1. : System Overview

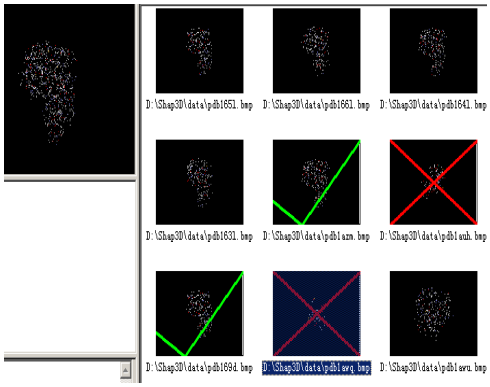

Fig. 2. User Interaction

The paper is organized as follows. Section 2 describes feature extraction and matching technology. Section 3 gives our relevance feedback method. Experimental results and conclusion are given in section 4 and section 5 respectively.

2. FEATURE EXTRACTION

First of all, we must find the best alignment of two proteins before we extract the features for matching. All the chemical atoms of the protein are treated as equally weighted points in our algorithm. The center of mass of each protein therefore is easily computed and then the protein can be centered at the origin. This method is commonly used to find center of the mass before alignment [6]. After that, in order to align the 3D protein structures which can rotate freely, a 3x3 matrix C is constructed by the covariance of the three dimensional model:

Assume $A = [x, y, z]^T$ and Ai is the (x,y,z) coordinates of each atom. The following formula calculates the covariance matrix C.

$$\mu = \frac{1}{M}\sum_{i=1}^{M} A_i \qquad C = \frac{1}{M}\sum_{i=1}^{M} [A_i - \mu][A_i - \mu]^T \tag{1}$$

The principle axis is then obtained by computing the eigenvectors of matrix C, which is also known as the principle component analysis (PCA). The eigenvector corresponding to the largest eigenvalue is the first principal axis. The next eigenvector corresponding to the secondary eigenvalue is the second principal axis, and so on. The two proteins will then be rotated the atom sets to their own principal axes for best alignment. PCA-based protein 3D alignment can make the two proteins to be rotated

to their maximal various directions. Then we need not consider the effect of different direction rotation of each protein in the following feature extraction.

After alignment, we extract geometry-based features of a set of points combined with protein description features before performing matching and retrieval. In classical mechanics and statistical theory, the concept of moments is used extensively [6]. We can extract nine features of each protein: the number of atoms, the weight of atoms, the render scale, two aspect ratios defined by height and depth divided by width, and four 2nd and 3rd order moments including $M_{200}, M_{210}, M_{102}, M_{201}$. Those nine features have been regarded as the most crucial features in protein structural description [6]. Then we can assign each feature a weight. The next section will give the method how to adjust the weight to make the protein retrieval standard more consistent with biologist judgment.

3. RELEVANCE FEEDBACK

For most previous retrieval systems (such as [8,9,10]), the retrieval processing follows steps below: Firstly, each object is represented by a set of selected features. Secondly, during the retrieval process, the user provides a query example or a set of feature values representing the query object and specifies a weight for each feature to assess the similarity between the query object and objects in the database. Finally, based on the selected features and specified weights, the retrieval system tries to find object similar to the user's query. The weights are fixed during the retrieval process.

Definition 1 (distance between two object)

Each object O is represented by a feature vector $f = [f_1,...,f_n]$ (n is the dimension of the feature). The distance between two objects is defined as

$$dist\ (f_j, f_k, w) == \left[\sum_{i=1}^{n} w_i \left| f_{ji} - f_{ki} \right|^2\right]^{1/2} \tag{2}$$

We assume that the weights are normalized. As we have seen, for most retrieval systems, the weights are fixed. The query model could not adapted to user's intention, which is often difficult to specify for users. To better capture the right weights reflecting users intention, relevance feedback is introduced in our system. The system provides an interface to gather user's evaluating information and learn from the positive and negative examples submitted by the user. Then the system computes the weights. The technique is called re-weighting. Rui [8] proposed a global re-weighting technique. We find that if positive examples are not consistent in some features, the efficiency of Rui's re-weighting method is very low. We propose a relevance feedback approach based on subspace to improve the retrieval efficiency.

3.1 Subspace Based Relevance Feedback Retrieval

The detail of the processing is shown as following steps.

1.At the initial retrieval time, the system computes the distance from query example to each object according to definition 1 and ranks the objects according to the distance value. The most similar object is ranked at the first and the minor is at the second.

2.If user is not satisfied with the retrieval results, he can give some positive examples and negative example.

3.The system clusters these positive examples into several subspaces, and creates some exceptions according to the negative examples.

4.The system computes the distances according to the new retrieval models and display retrieval results.

5.Repeat step 2 to 4, until user is satisfied the retrieval result.

3.2 Subspace Definition

Subspaces are obtained by clustered the training example sets. The training example sets are composed of query examples and positive examples supposed in relevance feedback retrieval. We partition these features of training examples into several subspaces.

A subspace represents a set of features that are similar in feature space. Each subspace $S =< m, C, R, W, P >$ consists of a set of weights $W = \{w_1,..., w_n\}$, a set of training examples $P = \{p_1,..., p_m\}$ that are clustered into the subspace, a centroid C that is the center of P and a radius that repents the mean distance from training examples of the subspace to centroid C. m is the number of training examples clustered into the subspace.

After each feedback, the system should compute distances of examples from subspaces. If the distance is less than or approximately equal to the radius the subspace, the new example is added to the subspace. . When a new example is added, it is either merged into an existing subspace or starts off a new subspace. If the number of subspace exceed $N_{sub\_\max}$, certain two subspaces are merged into one subspace. The nearest two subspaces are merged according to following distance.

$$dist\ (S_j, S_k) = \frac{n_j}{n_j + n_k} * dist\ (C_j, C_k, W_j) + \frac{n_k}{n_k + n_j} * dist\ (C_k, C_j, W_k) \quad (3)$$

Either adding new examples to subspace or merging two subspaces, the related parameters are set again. We use the re-weight method proposed by Rui [14]. After re-weighting and computing new centroid, the radius is changed according to the mean distance from every example to centroid.

3.3 Exception Definition

It is obviously that using reasonable negative information can improve the performance of system. But superabundant negative information will destroy the query model and degrade the efficiency of retrieval [15]. For most systems, the abilities to process negative information are finite.

In general, negative examples' are similar to query object in term of feature. Lacking of effective negative information processing ability will not only lose a part of feedback information, but also affect the retrieval efficiency.

Negative examples are regarded as exception. In other word, they are not selected as train examples to create subspace. On the contrary, we create many small negative example clusters that called exceptions.

Each exception $E =< m, C, R, W, N >$ consists of a set of weights $W = \{w_1,..., w_n\}$, a set of negative examples $N = \{n_1,..., n_m\}$ that clustered into the exception, a centroid C that is the center of N and a radius that repents the mean distance from negative examples of the exception to centroid C. m is the number of negative examples clustered into the exception.

After each feedback, the exceptions are updated according to the new negative examples provided by the user. The distances from negative example to each exception are computed. If the distance is larger than the radius of the exception, a new exception is created.

3.4 Similarity Definition

Through above-mentioned process, the positive examples are clustered into several clusters. These clusters represent the distribution of query objects. The retrieval process converts to compute from each object to these subspaces.

The radii of each subspace are different, which will affect the similarity judgment of the system. There are need a normalization schema. We define a sort of normalization method, which uses subspace radius to adjust the distance as following.

$$Dist'_k(O) = \begin{cases} 1 - \dfrac{1 - Dist_k(f, C_k, W_k)}{1 - R_k}, & if \quad Dist_k(o) >= R_k \\ 0, & if \quad Dist_k(o) < R_k \end{cases} \tag{4}$$

$Dist_k(f, C_k, W_k)$ is the distance from object O to subspace S_k. R_k, C_k and W_k are parameters related to subspace S_k. If an object is similar to the centroid of certain subspace, the object is a good result. The similarity from object O to query is defined as following.

$$Sim(O, Q) = \max(1 - Dist'_k(O)), \; k = 1,..., N_{sub} \tag{5}$$

Sometimes the centroid of exception is very near to the centroid of certain subspace. If the distance from object to certain exception is smaller than the radii of the exception, it is considered that $Sim(O, Q) = 0$.

4. EXPERIMENT

The experiments are conducted on the public database PDB (Protein Data Bank) [13], which includes 18691 protein files (As September 10, 2002) and FSSP database [16], in which the fold classification is based on structure-structure alignment of proteins,

and which includes all protein chains (>30 residues) from the PDB. FSSP database is based on search results of Dali engine (mainly based on the Z-score) and divides proteins in PDB into families. Each protein family has a representative protein. At the same time some of the results in FSSP have been revised by biologists. The FSSP database we used has 2860 protein families, which represent 27181 protein structures.

Since there is no universal agreement on the similarity of proteins and FSSP database is relatively objective and maintained by biologists, we use the FSSP classification results as the ground truth to make assessment of our retrieval system. The results are showed in Table 1. In the table, Representative Protein ID means the ID (entry) of the representative protein from each protein family in FSSP database. The Retrieval Accuracy is computed as follows: choose N (in our experiments, N=20) proteins whose similarity scores are ranked top in all proteins in PDB. Find M proteins that also belong to the same family as the representative protein in FSSP. Then the retrieval accuracy is simply M/N. As we can see, the first and second rounds of relevant feedback always improve the retrieval accuracy, and the ratio of total improvement is about 15.5% on average. This improvement is quite significant compared to the improvements obtained in some previous work [6].

Table 1. Retrieval Accuracy of different representative sets

Representative Protein ID	Initial Matching (Retrieval Accuracy)	First Relevant Feedback (Retrieval Accuracy)	Second Relevant Feedback (Retrieval Accuracy)
1cxq	80%	90%	100%
1a6m	80%	90%	95%
1abw	90%	100%	
1ba1	85%	100%	
1b8j	70%	75%	75%
1ctq	45%	65%	75%
1djx	40%	45%	55%
1fnc	35%	40%	40%
1crb	20%	25%	30%
1ckq	30%	35%	40%

5. CONCLUSION

In this paper, an intelligent protein 3D structure retrieval system is proposed. It is a novel matching method, which not only obtains 3D features with the moment technology, but also refines the query and the similarity metric in the retrieval process with relevant feedback. Since there is no universal agreement on the similarity of proteins, the major advantage of our system compared to other previous systems is that the relevance feedback technology is adapted to improve the similarity metric formula dynamically with biologist's interaction. The experimental results show that

the proposed approach can capture the biologist's intention and obtain significant improvement in the protein 3D structure retrieval.

6. ACKNOWLEDGMENTS

This research is supported by National Key Basic Research & Development Program. (No. 2002CB713807)

7. REFERENCES

[1] Singh, A. P. and Brutlag, D. L. (2001). Protein Structure Alignment: A comparison of methods. Nature Structural Biology, Submitted.
[2] http://gene.stanford.edu/3dSearch
[3] Holm L, Sander C (1993). Protein structure comparison by alignment of distance matrices. J Mol Biol 233, 123-138.
[4] Tsigelny1 I., Shindyalov I. N., Bourne P. E., Südhof T. and Taylor P.(2000) Common EF-hand Motifs in Cholinesterases and Neuroligins Suggest a Role for CA2+ Binding in Cell Surface Associations. Protein Science 9(1) 180-185.
[5] C. Zhang and T. Chen, "Efficient Feature Extraction for 2D/3D Objects in Mesh Representation", ICIP, 2001.
[6] Shann-Ching Chen and Tsuhan Chen, Retrival of 3D protein structures", ICIP, 2002.
[7] Ingvar Eidhammer, Inge Jonassen, William R. Taylor, Structure Comparison and Structure Patterns, Reports in Informatics No. 174, Department of Informatics, University of Bergen, Norway, July 1999.
[8] Yong Rui, Thomas S. Huang, Sharad Mehrotra, and Michael Ortega. Relevance Feedback: a power tool for interactive content-based object retrieval. IEEE trans. Circuits and systems for video technology, vol. 8, no. 5, pp. 644-655, Sep. 1998.
[9] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. Mindreaner: Query Databases Through Multiple Examples. In Proc. of the 24th VLDB Conference, (New York), 1998.
[10] Catherine. Lee, Wei-Ying Ma, Hongjiang Zhang. Information Embedding Based on User's Relevance Feedback for Object Retrieval. Technical report HP Labs, 1998.
[11] Nuno Vasconcelos and Andrew Lippman. Bayesian Representations and Learning Mechanisms for Content Based Object Retrieval. SPIE Storage and Retrieval for Media Databases 2000, San Jose, California, 2000.
[12] Lijuan Duan Wen Gao Jiyong Ma. Rich Get Richer Strategy for Content-Based Image Retrieval, Robert Laurini (Ed.): Advances in Visual Information Systems, 4th International Conference, VISUAL 2000, Lyon, France, November 2-4, 2000, Proceedings. Lecture Notes in Computer Science 1929 Springer 2000, pp290-299.
[13] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne: The Protein Data Bank. Nucleic Acids Research, 28 pp. 235-242 ,2000
[14] Y. Rui, T. S. Huang, A novel relevance feedback technique in image retrieval, ACM Mulitmedia 1999.
[15] J. J. Rocchio, Relevance feedback in information retrieval, In The SMART Retrieval System, Experiments in Automatic Document Processing, Pages 313-323. Prentice Hall, Engle-wood Cliffs, New Jersey, USA, 1971.
[16] L. Holm and C. Sander , Mapping the protein universe. Science 273:595-602, 1996.