Structural Building Blocks

Construction of Protein 3-D Structures Using a Structural Variant of Mountain Clustering Method

BY KEN-LI LIN, CHIN-TENG LIN, NIKHIL R. PAL, AND SUDEEPTA OJHA

We propose a modified version of the mountain clustering method (MCM) to find a library of structural building blocks for the construction of three-dimensional (3-D) structures of proteins. The algorithm decides on building blocks based on a measure of local density of structural patterns. We tested our algorithm on a well-known data set and found it to successfully reconstruct a set of 71 test proteins (up to first 60 residues as done by others) with lower global-fit root mean square (RMS) errors compared to an existing method that inspired our algorithm. The constructed library of building blocks is also evaluated using some other benchmark data set for comparison. Our algorithm achieved good local-fit RMS errors, indicating that these building blocks can model the nearby fragments quite accurately. In this context, we have proposed two alternative ways to compare the quality of such quantization and reconstruction results, which can be used in other applications too.

Discovering the relations between protein sequences and their 3-D structures is an important research topic and has received a lot of attention, because knowing the 3-D structure of a protein helps biologists to study the functions of the proteins, perform rational drug design, and design novel proteins. Finding the 3-D structure of a protein by using X-ray crystallography or nuclear magnetic resonance imaging is time consuming and expensive; hence, alternative approaches are being tried. Several approaches such as comparative modeling, fold recognition [1], [2], ab initio prediction [3], [4], and 3-D building blocks approaches [5]–[8] have been proposed. As pointed out by Bujnicki [7], modeling of a protein structure de novo without using templates is quite difficult because the search space is enormous even for proteins of moderate sequence lengths. The methods based on the assembly of short fragments have shown a great promise [5]–[18]. Among these methods, 3-D building blocks approaches have been successfully applied to construct libraries of well-chosen short structural motifs extracted from known structures. These building blocks are then used to construct or analyze structures of new proteins. The short structural fragments that recur across different protein families can often be viewed as stand-alone units that fold independently and hence can help assignment of building blocks to unknown proteins for reconstruction of 3-D structures [15].

Digital Object Identifier 10.1109/MEMB.2009.932912

The clustering method used in [15] is a two-stage process, where building blocks are classified according to their structural classification of proteins (SCOP) family and clustered within the family in the first stage and then merged in the second stage. The building block cutting algorithm uses a stability score function that involves properties like compactness, hydrophobicity, and isolatedness. The critical building block finding algorithm uses a score function based on the contacts the building block has with other building blocks. This is an involved and interesting approach. Our proposed approach is comparatively very simple and does not use the physical or chemical or structural properties of the residues.

In [16], Anishetty et al. suggested that rigid tripeptides have no correlation with the protein's secondary structure, and tripeptide data may be used to predict the plausible structures for oligopeptides. The hybrid protein model of de Brevern et al. obtains 3-D protein fragments encoded into a structural alphabet consisting of 16 protein blocks (PBs) [18], [19]. Benros et al. [17] further continued this study considering 11-residue fragments encoded as a series of seven PBs. They had built a library of 120 overlapping prototypes with good local approximation of 3-D structures. Every PB in [18] is only five-residue long and described by eight dihedral angles. Each of them serves as a building block approximately representing a known structural motif, such as central α-helices, central β-strands, and β-strand-N-caps. Consequently, a protein's 3-D structure can be represented by a string of alphabets. Unlike our approach, the similarity between fragments is defined by the RMS deviation (RMSD) on angular values, and the clustering algorithm used is a self-organizing map-type neural network.

The effectiveness of such a method heavily depends on the extraction of good representative 3-D building blocks. Unger et al. [5] proposed a two-stage clustering algorithm (TSCA) to choose hexamers having a large number of neighbors to be the building blocks. These center hexamers are called the 3-D building blocks [5]. Micheletti et al. also used the largest number of nearby points within a similarity cutoff called *proximity score* [6] to select cluster centers, while Kolodny et al. proposed a simulated annealing k-means method to perform the clustering task with the minimal total variance score [8], [13]–[14]. In this article, a modified form of the MCM or subtractive clustering method (SCM) [20], [21] is proposed to find building blocks. Our experiments with some

 0739-5175/09/$25.00©2009IEEE

benchmark data sets show that it can find better representative building blocks than the method in [5]. We also propose two alternative ways of depicting the quality of the building blocks.

3-D Building Block Approaches

The 3-D building block approach involves several steps. First, we need to decide on the fragment length. Then, given a set of proteins (training data), we need to compile the whole set into all possible fragments of the specified length. Next, a clustering method is used to divide these fragments into clusters and pick up the center of each cluster to be a building block. If these building blocks are good enough, then they can be used successfully to represent all original fragments within a tolerable limit and therefore can be used to reconstruct the 3-D structure of a whole protein within some tolerance.

Distance Measure Between 3-D Structures

A well-accepted definition of dissimilarity between two fragments is the RMSD between two structures computed after alignment of the two fragments to the greatest possible extent using the best molecular fit (BMF) algorithm [22], [23]. Given two structures s and t, the RMS can be calculated as follows:

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^{K} \left\| r_i^s - r_i^t \right\|^2}{K-2}}, \tag{1}$$

where r_i^s is a 3-D coordinate of ith C_α atom in the molecule s and K denotes the number of atoms in the structure. Typically, for the computation of RMS, one should divide by K, but for the ease of comparison with published results, we divide by $(K-2)$ as done in [5].

Method of Reconstruction

Following [5], we use this procedure: first, we replace each original hexamer of a protein by its closest building block. Then, since the building blocks overlap, we align every two consecutive building blocks by using the BMF algorithm. The chain grows as follows. Onto the suffix (the last five residues) of the first building block, we fit the prefix (the first five residues) of the next building block. The 3-D position of the sixth (last) residue of the latter hexamer is thus determined and is added to the growing chain. This process is repeated until the whole protein is reconstructed.

Performance Measure

To evaluate the performance of the proposed method, we use the same two criteria as in [8]: 1) local-fit RMS, which measures how well the fragments of the target proteins can be represented by the library of building blocks at hand. It takes an average of all coordinate RMSDs between every fragment and its associated building block. 2) Global-fit RMS, which measures the RMSD of the reconstructed 3-D structure from the entire native structure of the target. In addition, we also use two alternative ways, as explained later, for assessment of quality of the building blocks.

Two-Stage Clustering Algorithm

Since we will compare our results with those by the algorithm in [5], we briefly describe the same. The TSCA defines a cluster as a set of structures such that the RMSD of any member in the cluster from a designated representative member is less than a threshold. In [5], 1 Å is used as the separation between similar and dissimilar hexamers and hence as the threshold for defining clusters. In the first stage, a randomly chosen hexamer is taken as the center of the first cluster, and all hexamers which are within 1 Å distance after the best molecular fit are placed in that cluster. Each member of this cluster then acts as a new center and adds all of its neighbors which are within the threshold. This annexation process is continued till no more hexamers can be added to the cluster. Then, another unused hexamer is taken as the center of the next cluster, and the process is repeated to get the next cluster. The entire process is repeated till every hexamer is included in some cluster. It is obvious that, in such a cluster, the maximum distance between a pair of hexamers could be much higher than 1 Å. In the second stage, these big clusters are divided into smaller clusters such that every member of a cluster is within a distance of 1 Å from a centroid. For each cluster, the hexamer with the maximum number of neighbors within 1 Å is taken as the center of a new subcluster having those neighbors as members. The process is repeated until all hexamers of the cluster are assigned to subclusters.

MCM and SCM

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \Re^p$ be a set of n data points in p dimensions. We denote x_{jk} as the jth component of the kth point $\mathbf{x}_k$; $k = 1, 2, \ldots, n$; $j = 1, 2, \ldots, p$. The MCM [20] generates a set of N equispaced grid points $\mathbf{v}_i$, $i = 1, 2, \ldots, N$ in $\Re^p$ over the smallest hypercube (in $\Re^p$) containing X. Then, at every grid point, a potential value (called mountain potential) is computed, which represents a kind of local density of points around the grid point. Now the grid point with the maximum mountain potential is selected as the first cluster center. To find other cluster centers, the mountain function values are discounted to reduce the effect of already detected centers, and the grid point corresponding to the highest peak of the discounted potential is taken as the next cluster center. This process of discounting and finding of cluster centers is continued until the discounted potential becomes too small to look for useful clusters.

In an MCM, the quality of the centers depends on the fineness of the grid, and better resolution leads to more cost. The computational overhead increases rapidly with dimension p. To reduce the computational overhead of MCM, Chiu [21] suggested a modification of MCM, known as the SCM.

Instead of imposing artificial grids, Chiu [21] used each data point as a potential cluster center. Following the MCM, the potential function is defined as

$$P_1(\mathbf{x}_i) = \sum_{k=1}^{n} e^{-\alpha d^2(\mathbf{x}_k, \mathbf{x}_i)}; \quad i = 1, 2, \ldots, n, \tag{2}$$

and discounting the potential on subsequent steps is done as follows:

$$P_k(\mathbf{x}_i) = P_{k-1}(\mathbf{x}_i) - P_{k-1}^* e^{-\beta d^2(\mathbf{x}_{k-1}^*, \mathbf{x}_i)} \tag{3}$$

for $k = 2, 3, \ldots, c$ and $i = 1, 2, \ldots, n$. Here $\mathbf{x}_{k-1}^*$ is the $(k-1)$th (most recently detected) cluster center, and α and β are positive constants. The rest of the SCM algorithm remains the same as that of the mountain method. Unlike MCM, here the number of prospective cluster centers is n, and hence is independent on the dimensionality and spread of the data. Chiu [21] terminated SCM when $P_{k-1}^*/P_1^* < \delta$, $0.0 < \delta < 1.0$. Although,

Algorithm 1: Structural mountain clustering

Begin

Input data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \Re^p$

Choose α

Compute $d(\mathbf{x}_i, \mathbf{x}_j) = \text{RMS}_{i,j}$, for all $i, j = 1, 2, \ldots, n$ using the BMF algorithm; $\text{RMS}_{i,j}$ is the root mean square distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ after BMF.

Repeat while any hexamer is left to be assigned to a cluster

Calculate the potential at each hexamer $\mathbf{x}_i$ using (2)

Find the hexamer with the highest potential and choose it as a building block

Remove all hexamers, which are within a RMS error of 1 Å from the building block, to form the cluster associated with the building block.

Continue

End Repeat

SCM reduces the computational complexity, it will give good results only if the desired cluster centers (points corresponding to the maximum local density) coincide with one of the data points or close to it. For the present problem, since we have to choose one of the hexamers as the center, the SCM framework is quite suitable.

Structural Mountain Clustering Methods

This is a modified form of subtractive MCM [20], [21] to handle structural data such as hexamers. For hexamers, the use of Euclidean distance will not be meaningful, because the Euclidean distance between the two hexamers, where one is a translated version of the other or one is a rotated version of the other, would be high while for our purpose they are the same. Suppose the set of hexamers is represented by $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \Re^p$. In structural mountain clustering methods (SMCMs), each hexamer is considered a potential cluster center. Instead of the Euclidean distance, the contribution made by a hexamer $\mathbf{x}_j$ to the potential associated with another hexamer $\mathbf{x}_i$, $i \neq j$, depends on the structural similarity between $\mathbf{x}_j$ and $\mathbf{x}_i$. The structural similarity is obtained after aligning the data points using the BMF routine [22], [23]. Thus, the higher the similarity between the two hexamers, the more quantity is added to the potential. In this way, at every hexamer, we compute the mountain potential P using all other hexamers. After this, like MCM we find the hexamer, $\mathbf{x}_k$, with the highest potential as the first building block. Then, we form the first cluster, taking all hexamers that are within 1 Å of RMS after the best molecular fit.

We now remove all members in the first cluster and recompute the potential to find the next cluster center. Note that MCM and SCM neither remove any data point nor recompute the potential. Here we recompute the potential as we want every cluster center to be at the center of a dense region. To get the third cluster, the members of the second clusters are removed and the potential is recomputed. The process is continued until every data point is assigned to some cluster as described in the Algorithm 1.

The choice of α may have an effect on the clusters extracted, and hence, we experimented with different choices of α to get an optimal value for it.

SMCM Can Produce Better Building Blocks than TSCA

Note that to find a local estimate of the density, SMCM takes into account the geometry of the data and not just the count of number of points within a cutoff distance and, hence, it is likely to produce better building blocks. For example, consider a two-dimensional data set having 31 points, such that one point is at the center of a circle of radius 1 Å and the remaining 30 points are grouped into two clusters, each having 15 points such that 10 points from each cluster are within a distance of 1 Å from the central point. Here, TSCA will take the center point as a building block as it will have 21 points (including itself) within 1 Å, while the remaining 5 points from each cluster will form two other clusters. Clearly, these clusters and building blocks are not the desirable ones. But the SMCM will identify the center of each cluster with 15 points as the building block. These building blocks are better than those selected by TSCA, because SMCM building blocks are at the centers of dense regions. The isolated central point will also be extracted as a building block, but since it is supported by only one point, it is a poor building block and can be discarded. The SMCM is also expected to find better representative building blocks than hierarchical clustering or k-means type clustering. This is so because hierarchical clustering algorithms do not pay attention to the density of points (here density of similar structures). Moreover, a

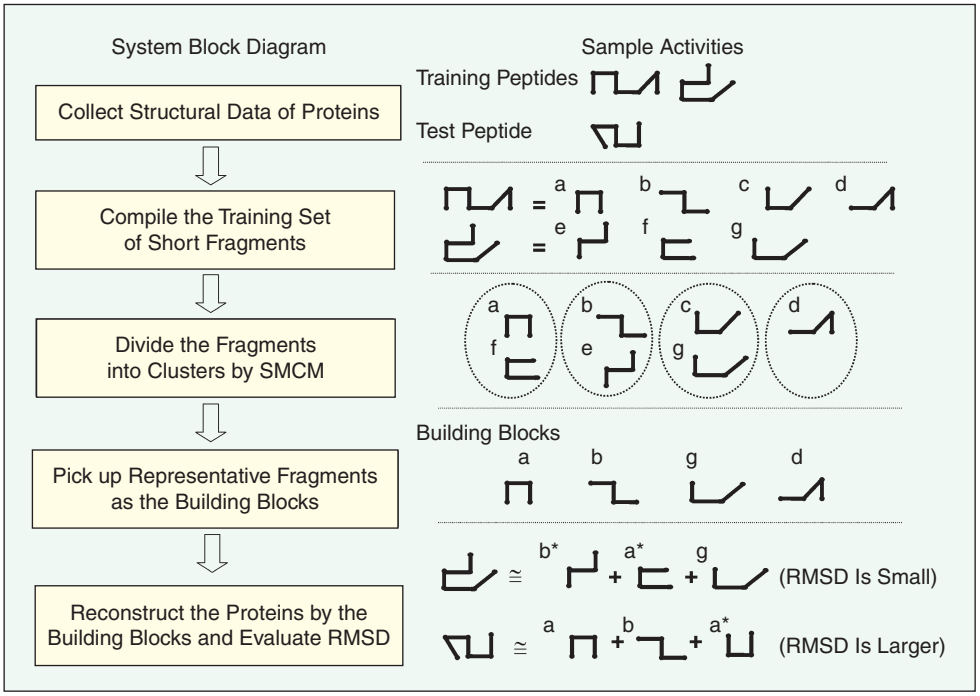

Fig. 1. A block diagram of the building block approach using SMCM. Note that fragment **x*** represents the fragment **x** after some rotation and alignment.

The effectiveness of the method heavily depends on the extraction of good representative 3-D building blocks.

hierarchical clustering algorithm does not produce any prototypical building blocks. The poor performance of hierarchical clustering algorithms for fragment data is also pointed out in [8]. The usual *k*-means type clustering is also not very appropriate for such a problem, as the mean of a set of 3-D structures (even after best alignment) will not have any associated residue sequence and hence is difficult to interpret.

For an easy understanding of the entire process, on the left part of Figure 1, we provide a block diagram and on the right side of each block we illustrate the activity in the block by using a simple data set consisting of two training peptides and one test peptide. For illustration, we consider fragments of length four. Thus, the two training peptides result in seven fragments named **a**–**g** in Figure 1. In the third step (block), the SMCM finds four clusters. For example, in the first cluster, fragments **f** and **a** are placed together because they are almost the same after BMF alignment. The four clusters result in four building blocks, **a, b, g,** and **d**, as shown in the fourth block on the right part of Figure 1. In the last step, we show two reconstruction cases of which the first one, (**b***+**a***+**g**), has a smaller reconstruction error than the second one, since its corresponding building blocks are better matched. It is noted that **b*** represents the building block **b** after alignment.

Data Sets

We have used the same 82 peptides as in [5]. (The list of peptides used can be found in Table I of [5].) To create the library of fragments, only the C_α coordinates are used. We use the same four proteins (1BP2, 1PCY, 4HHBb, 5PTI) as in [5] as the training set. The data in the Protein Data Bank (PDB) are updated as new information becomes available. As of December 2006, in PDB, the information about the following 21 proteins was changed: 1APR (2APR), 1CPP (2CPP), 1CPV (5CPV), 1FB4h (2FB4h), 1FBJl (2FBJl), 1FDX (1DUR), 1GCR (4GCR), 1HMQa (2HMQa), 1INSa (4INSa), 1PCY (1PLC), 1SN3 (2SN3), 3PTP (5PTP), 3RXN (7RXN), 3TLN (8TLN), 4ADH (8ADH), 4ATCa (6AT1a), 1GAPa (1G6Na), 2FD1 (5FD1), 4FXN (2FOX), 2APP (3APP), 4CYTr (5CYTr). The new PDB ID is shown within parentheses. We have used both the old database as used in [5] and the new database as of December 2006.

Experimental Results

The SMCM has only one parameter α. Using fragments of length 6, we have experimented with different choices of α such as $\alpha = 2, 3, 4, 5,$ and 6, using the same database as used in [5]. We have found $\alpha = 4$ and 5 to yield better results. So, we further fine-tuned α in the range of 4–6 in steps of 0.5. Finally, we have got $\alpha = 5$ to produce the best result with a global-fit RMS 7.19, which is less than 7.3 than is reported in [5]. We have also experimented with the newly updated database. For the new data set, $\alpha = 5.5$ resulted in the best global-fit RMS of 7.32. Table 1 summarizes the variations in the local-fit RMS error and global-fit RMS error with the same choice of α for both data sets.

To further compare the quality of building blocks, we have implemented the TSCA method on the same data. We have obtained 55 main clusters and 102 subclusters (Unger et al. reported 103). SMCM extracted 104 building blocks. So for a fair comparison, we remove the trailing two building blocks from 104 building blocks. Thus, for both methods, we use the same number of building blocks to represent all target fragments and reconstruct the first 60 residues of 71 proteins whose lengths are larger than 60 residues using the same approach as in [5]. For our method, when we use only 102 clusters, the local-fit RMS increases to 0.75 and global-fit RMS increases to 7.23, which is still better than 7.3. Our implementation of TSCA results in a local-fit RMS of 0.77 and a global-fit RMS of 8.27, and these are higher than the values reported in [5]. Since the clustering algorithm is an iterative one, we provide an approximate analysis of its complexity. Note that, although the repeat–end repeat loop in the algorithm uses (2), $e^{-\alpha d^2(\mathbf{x}_k, \mathbf{x}_i)}, \forall\ i, k$ can be computed just once for all and stored in a table to be used in the repeat loop. Let us assume that the cost for computing $e^{-\alpha d^2(\mathbf{x}_k, \mathbf{x}_i)}$ for any pair $\{\mathbf{x}_k, \mathbf{x}_i\}$ be D. D involves the cost of computing the RMSD using the best molecular alignment of two segments, the cost of a multiplication, and

Table 1. Effect of the choice of α on the local-fit RMS error and the global-fit RMS error for the SMCM algorithm when the fragment length is 6.

α	Library Size	Local-Fit RMS	Global-Fit RMS
Original data set			
6	106	0.742	7.64
5.5	106	0.742	7.64
5	104	0.749	7.19
4.5	104	0.750	7.27
4	105	0.748	7.30
3.5	104	0.750	7.62
3	103	0.751	7.92
2	105	0.746	7.95
Newly updated data set			
8	108	0.726	7.53
7	107	0.727	7.48
6.5	107	0.727	7.48
6	107	0.723	7.32
5.5	107	0.723	7.32
5	107	0.725	7.59
4.5	107	0.727	7.69
4	106	0.728	7.75

that of computing the exponential. Thus, the total cost of computing $e^{-\alpha d^2(\mathbf{x}_k, \mathbf{x}_i)}$, $\forall\, i, k$ is $n(n-1)D/2$. The iterative part of the algorithm needs to make additions of such values read from the table. Let us assume that n hexamers are iteratively divided into c clusters and each cluster is of size k on an average, i.e., $n = ck$. In the first iteration, the potential for each hexamer is calculated by the summation of n exponential values taken from the table. It requires n^2 additions for n hexamers. In the second iteration, since k hexamers are removed and assigned to the first cluster, the remaining hexamers require $(n-k)^2$ additions in the computation of potential. Likewise, in the final iteration, it requires k^2 additions. So the required time in addition operation to compute potential for the entire algorithm is

$$\sum_{i=0}^{c-1} (n-ik)^2\, A = k^2\, \frac{c(c+1)(2c+1)}{6} A. \qquad (4)$$

In (4), A is the cost of one table lookup and that of an addition operation. The total computation time required is thus $T = n(n-1)D/2 + k^2\, c(c+1)(2c+1)A/6$. Since A is a constant and D is assumed to be a constant, the first term in T is of $O(n^2)$ and the second term is of $O(n^2 c)$. For this specific data set, SMCM takes 2 min 48 s on a personal computer with a 3.4-GHz CPU and 1-GB RAM to find the clusters, whereas TSCA uses 51 s for the clustering task. One reason for the noticeable difference in the running time between the two algorithms is that we did not use the efficient table-lookup procedure but evaluated the distances and exponentials in every iteration of the repeat-loop in the algorithm. The use of small structural motifs as building blocks is based on the hypothesis that there are repeatedly occurring stable fragments and, hence, in practice we shall not need to run such an algorithm on a very large database.

To have a visual assessment of the quality of the building blocks and the reconstruction [Figure 2(a)–(c)], we depict one of the most frequently used building block (NCYKQA), a very good fit target hexamer (LANWMC), and its representation using the building block. Figure 2(a) is the original building block (in solid red lines), while Figure 2(b) shows the original target hexamer (in dashed blue lines). Although at first sight the two structures look quite different, after the best alignment, the superimposed structures look identical [Figure 2(c)]. It is interesting to observe that the local secondary structures of both fragments are all alpha helix. This indicates that the identified building blocks are biologically meaningful structural motifs. Figure 2(d) displays another building block (NKEHKN), and Figure 2(e) depicts a poor fit target fragment (AAHCKN, the RMS error is larger than previous example but is still less than 1 Å). In this case too, we find a very good match between the two structures in Figure 2(f).

The most typical helical building block found by SMCM is NCYKQA and is located at residues 50–55 of 1BP2; while the most-populated building block found by TSCA [5] is ICFSKV and it is located at residues 104–109 of 1BP2. From the fact that ICFSKV is also an all-helical structure and is included in the cluster of NCYKQA, it appears that the TSCA cluster associated with ICFSKV and the SMCM cluster associated with NCYKQA represent the same biological structural motif. Figure 3(a) and (b) shows these two building blocks, and it is clear that they represent the same structural unit. Similarly, we find that the most typical extended strand GKVTVN found by SMCM is located at residue 94 of 1PCY, while its counterpart NEITCS found by TSCA is located at residue 80 of 1BP2. These two building blocks are depicted in Figure 3(c) and (d). The hexamer CSSENN is another interesting building block found by SMCM. Unger et al. [5] pointed out that CSSENN represents a turn joining two beta strands. Thus, we find that building blocks found by SMCM represent structures of biological significance.

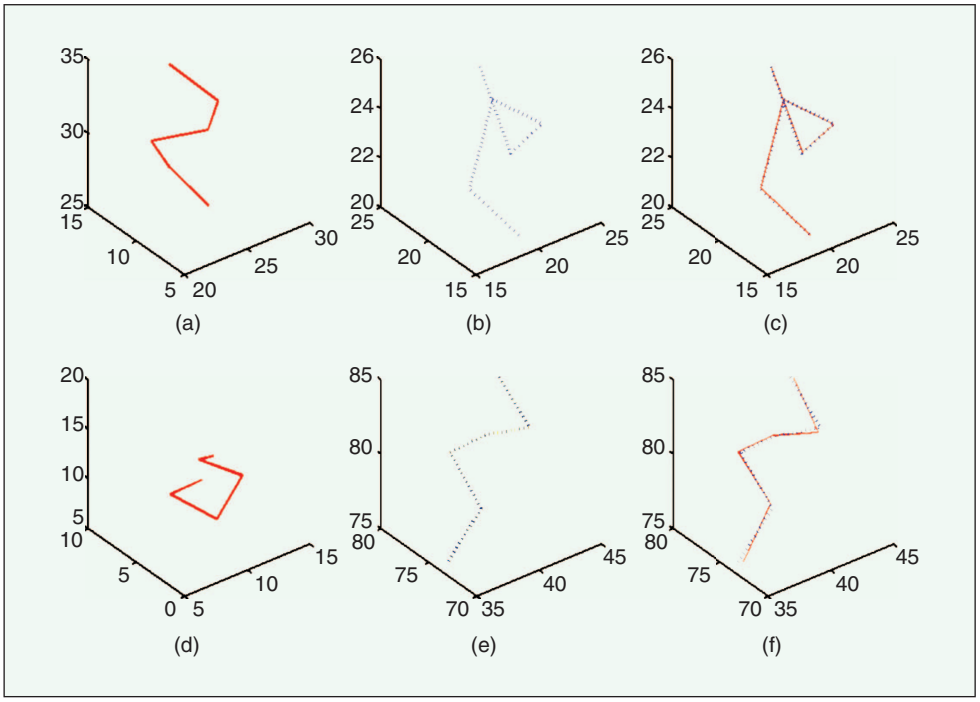

Fig. 2. Representation of target fragments using building blocks: (a) a building block with sequence NCYKQA; (b) a very good fit target hexamer with sequence LANWMC; (c) the building block and target hexamer superimposed after alignment (NCYKQA and LANWMC superimposed); (d) another building block with sequence NKEHKN; (e) a poor fit target hexamer with sequence AAHCKN; and (f) the building block and target hexamer superimposed after alignment (NKEHKN and AAHCKN superimposed).

Alternative Ways of Performance Evaluation

To evaluate the local-fit RMS quality, we compare the histograms of local-fit RMS error measuring the deviations of fragments from their corresponding building blocks [Figure 4(a)]. It shows that the total count of the lower RMS error (area under the curve) for SMCM is larger than that for TSCA, and this indicates that more fragments

are represented by good building blocks with lower reconstruction errors for SMCM.

Finally, we compare protein by protein the local-fit RMS error produced by SMCM and TSCA in Figure 4(b), where proteins are sorted in descending order of local-fit-RMS errors produced by SMCM. Figure 4(b) reveals that SMCM's local-fit RMS error is usually lower than the corresponding TSCA error. In this particular case, about 75% of the protein's SMCM local-fit RMS error is lower. The results on the updated database are found to be quite similar to the results on the original database. We have also experimented with fragment lengths 5–7 and found that, for the SMCM, the fragment length 6 is the optimal in terms of reconstruction error for both databases that we have experimented with. For SMCM, the reconstruction error obtained with fragment length 7 is 7.57, which is slightly higher than that with hexamers. However, for TSCA, the best reconstruction error of 7.59 is achieved with fragment length 7 while the error with fragment length 6 is 8.14.

Evaluation of the Library of Building Blocks on Other Data Sets

To evaluate the quality of the library of building blocks, we use it to reconstruct proteins used in two more recent studies by Micheletti et al. [6] and Kolodny et al. [8]. We have excluded a few proteins with sequence discontinuity [14]. In these two data sets, there are 10 and 144 proteins, respectively, which are used for testing. For the reconstruction, we follow a scheme similar in spirit with the method in [8]. The reconstruction process tries to minimize global-fit RMSD. Although reconstructing residue by residue, instead of using the building block with the best local-fit RMS, the one with the minimum global-fit RMSD is chosen. It is noted that local-fit RMSD at each residue during such reconstruction usually will be higher. For the Micheletti et al. data set, the global-fit RMSD obtained is 0.92 Å, which is slightly lower than 1.06 Å reported in [6]. For the data set used by Kolodny et al., authors reported the global-fit RMSD between 0.76 and 2.9 Å for different fragment lengths, whereas for this data set using hexamers, we have achieved a global-fit RMSD of only 1.05 Å, which is better than a global-fit RMSD of 1.26 Å reported in [8]. This further establishes that SMCM can extract biologically meaningful structural motifs that can be used for reconstruction of protein structures.

Conclusions

We proposed a modified form of MCM to deal with structural data and explained how this algorithm exploits the data geometry to yield better building blocks. We have applied this algorithm

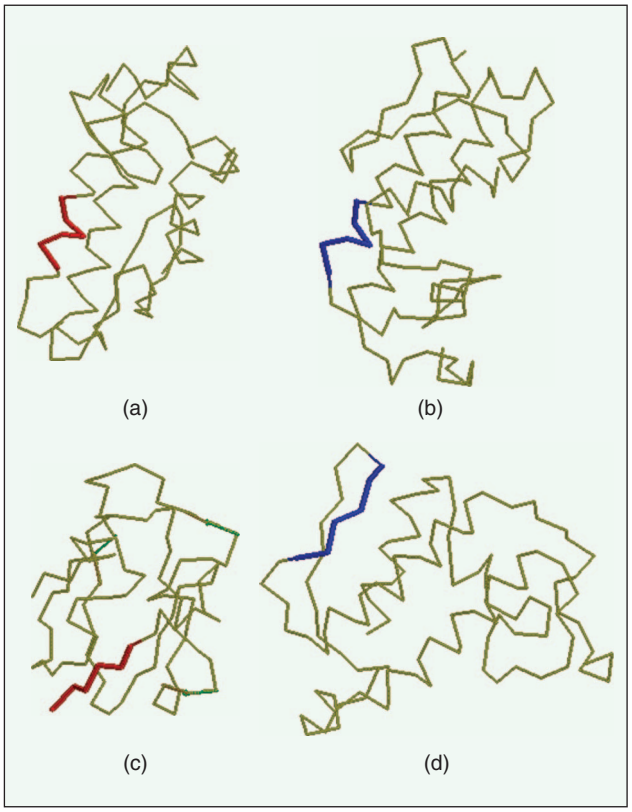

Fig. 3. (a) SMCM building block NCYKQA at residues 50–55 of 1BP2; (b) TSCA building block ICFSKV at residues 104–109 of 1BP2; (c) SMCM building block GKVTVN at residues 94–99 of 1PCY; (d) TSCA building block NEITCS at residues 80–85 of 1BP2.

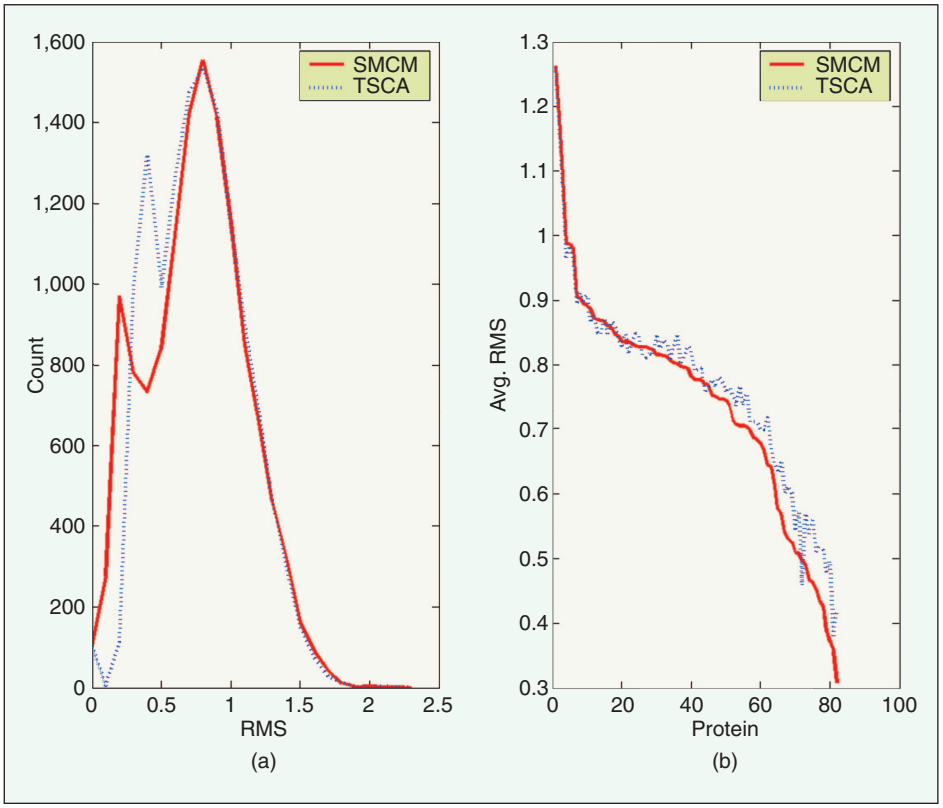

Fig. 4. (a) Histograms of local-fit RMS errors for SMCM and TSCA. (b) Protein-by-protein comparison of local-fit RMS error for SMCM and TSCA.

to find building blocks for reconstruction of protein's 3-D structures on the same data set used by Unger et al. and also on the most recent version of the same data set. These building blocks can successfully reconstruct the 3-D protein structure for the first 60 residues (as done by Unger et al.) of all test proteins with global-fit RMS error within 7.23 Å and also obtain good local-fit RMS error, indicating that they can model the nearby fragments within tolerable errors. Using the reconstruction error, we have demonstrated that SMCM building blocks are better than those obtained by TSCA. The superiority of the same set of building blocks found by SMCM is further established by reconstructing the 3-D structures of two additional data sets used by other researchers. The computation overhead of SMCM is also discussed. We have proposed two alternative ways to compare local-fit RMS errors, which reveal that the performance of SMCM building blocks is usually better than TSCA building blocks.

Acknowledgment

This work was supported by the National Science Council, Taiwan, Republic of China, under contracts NSC 97-2627-E-009-001 and NSC 97-2221-E-009-138, "Aiming for the Top University Plan," National Chiao Tung University (NCTU) and the Ministry of Education, Taiwan, under contract 97W806.

Ken-Li Lin received his B.S. and M.S. degrees in control engineering from NCTU, Taiwan, in 1990 and 1992, respectively. He is a Ph.D. candidate of the Department of Electrical and Control Engineering, NCTU, Hsinchu, Taiwan. He is currently with the computer center of Chung Hua University, Hsinchu, Taiwan, since 1998. His research interests include information technology, computational intelligence, and bioinformatics.

Chin-Teng Lin is currently the chair professor of electrical and computer engineering, the provost of the NCTU, and director of Brain Research Center at NCTU. He served as the dean of Computer Science College, director of the Research and Development Office of NCTU, the chair of Electrical and Control Engineering Department of NCTU, and associate dean of the College of Electrical Engineering and Computer Science. He has published more than 110 journal papers in the areas of neural networks, fuzzy systems, multimedia hardware/software, and soft computing, including about 90 IEEE journal papers. He currently serves as associate editors of *IEEE Transactions on Circuits and Systems, Parts I and II, IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Fuzzy Systems,* and *International Journal of Speech Technology.* He is a Fellow of the IEEE.

Nikhil R. Pal is a professor in the Electronics and Communication Sciences Unit of the Indian Statistical Institute. He has coauthored or edited and coedited several books. His current research interests include bioinformatics, medical and satellite image analysis, pattern recognition, fuzzy sets theory, neural networks, and evolutionary computation. He serves the editorial or advisory board of several journals. He is an associate editor of *IEEE Transactions on Systems Man and Cybernetics-B* and editor-in-chief of *IEEE Transactions on Fuzzy Systems*. He is a Fellow of the IEEE.

Sudeepta Ojha received his M.Tech. degree in computer science from the Indian Statistical Institute, Calcutta, in 2005. Currently, he is working with the Hong Kong and Shanghai Banking Corporation Ltd., India.

Address for Correspondence: Ken-Li Lin, Computer Center, Chung Hua University, No. 707, Sec. 2, WuFu Road, Hsinchu, Taiwan 300, Republic of China. E-mail: kennylin@chu.edu.tw.

References

[1] C. D. Huang, C. T. Lin, and N. R. Pal, "Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification," *IEEE Trans. Nanobiosci.*, vol. 2, no. 4, pp. 221–232, 2003.

[2] C. Bystroff and D. Baker, "Prediction of local structure in proteins using a library of sequence-structure motifs," *J. Mol. Biol.*, vol. 281, no. 3, pp. 565–577, 1998.

[3] Y. Liu and D. L. Beveridge, "Exploratory studies of ab-initio protein structure prediction: Multiple copy simulated annealing, amber energy functions, and a generalized born/solvent accessibility solvation model," *Proteins*, vol. 46, no. 1, pp. 128–146, 2002.

[4] P. Pokarowski, A. Kolinski, and J. Skolnick, "A minimal physically realistic protein-like lattice model: Designing an energy landscape that ensures all-or-none folding to a unique native state," *Biophys. J.*, vol. 84, no. 3, pp. 1518–1526, 2003.

[5] R. Unger, D. Harel, S. Wherland, and J. L. Sussman, "A 3D building blocks approach to analyzing and predicting structure of proteins," *Proteins*, vol. 5, no. 4, pp. 355–373, 1989.

[6] C. Micheletti, F. Seno, and A. Maritan, "Recurrent oligomers in proteins: An optimal scheme reconciling accurate and concise backbone representations in automated folding and design studies," *Proteins*, vol. 40, no. 4, pp. 662–674, 2000.

[7] J. M. Bujnicki, "Protein-structure prediction by recombination of fragments," *ChemBioChem*, vol. 7, no. 1, pp. 19–27, 2006.

[8] R. Kolodny, P. Koehl, L. Guibas, and M. Levitt, "Small libraries of protein fragments model native protein structures accurately," *J. Mol. Biol.*, vol. 323, no. 2, pp. 297–307, 2002.

[9] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker, "Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions," *J. Mol. Biol.*, vol. 268, no. 1, pp. 209–225, 1997.

[10] G. Chikenji, Y. Fujitsuka, and S. Takada, "A reversible fragment assembly method for de novo protein structure prediction," *J. Chem. Phys.*, vol. 119, no. 13, pp. 6895–6903, 2003.

[11] D. Kihara and J. Skolnick, "The PDB is a covering set of small protein structures," *J. Mol. Biol.*, vol. 334, no. 4, pp. 793–802, 2003.

[12] Y. Zhang and J. Skolnick, "The protein structure prediction problem could be solved using the current PDB library," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 4, pp. 1029–1034, 2005.

[13] R. Kolodny and M. Levitt, "Protein decoy assembly using short fragments under geometric constraints," *Biopolymers*, vol. 68, no. 3, pp. 278–285, 2003.

[14] B. H. Park and M. Levitt, "The complexity and accuracy of discrete state models of protein structure," *J. Mol. Biol.*, vol. 249, no. 2, pp. 493–507, 1995.

[15] N. Haspel, C. J. Tsai, H. Wolfson, and R. Nussinov, "Hierarchical protein folding pathways: A computational study of protein fragments," *Proteins*, vol. 51, no. 2, pp. 203–215, 2003.

[16] S. Anishetty, G. Pennathur, and R. Anishetty, "Tripeptide analysis of protein structures," *BMC Struct. Biol.*, vol. 2, no. 9, pp. 1–8, 2002.

[17] C. Benros, A. G. de Brevern, C. Etchebest, and S. Hazout, "Assessing a novel approach for predicting local 3D protein structures from sequence," *Proteins*, vol. 62, no. 4, pp. 865–880, 2006.

[18] A. G. de Brevern and S. Hazout, "Hybrid protein model for optimally defining 3D protein structure fragments," *Bioinformatics*, vol. 19, no. 3, pp. 345–353, 2003.

[19] A. G. de Brevern, C. Etchebest, and S. Hazout, "Bayesian probabilistic approach for prediction backbone structures in terms of protein blocks," *Proteins*, vol. 41, no. 3, pp. 271–287, 2000.

[20] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 8, pp. 1279–1284, 1994.

[21] S. L. Chiu, "Extracting fuzzy rules for pattern classification by cluster estimation," in *Proc. 6th Int. Fuzzy Systems Association, World Congress (IFSA'95)*, 1995, pp. 1–4.

[22] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallogr. A*, vol. 32, no. 5, pp. 922–923, 1976.

[23] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallogr. A*, vol. 34, no. 5, pp. 828–829, 1978.