Protein Clustering on a Grassmann Manifold

Chendra Hadi Suryanto[1], Hiroto Saigo[2], and Kazuhiro Fukui[1]

[1] Graduate School of Systems and Information Engineering,
Department of Computer Science, University of Tsukuba, Japan
http://www.cvlab.cs.tsukuba.ac.jp/
[2] Department of Bioscience and Bioinformatics,
Kyushu Institute of Technology, Japan
http://www.bio.kyutech.ac.jp/~saigo/

Abstract. We propose a new method for clustering 3D protein structures. In our method, the 3D structure of a protein is represented by a linear subspace, which is generated using PCA from the set of synthesized multi-view images of the protein. The similarity of two protein structures is then defined by the canonical angles between the corresponding subspaces. The merit of this approach is that we can avoid the difficulties of protein structure alignments because this similarity measure does not rely on the precise alignment and geometry of each alpha carbon atom. In this approach, we tackle the protein structure clustering problem by considering the set of subspaces corresponding to the various proteins. The clustering of subspaces with the same dimension is equivalent to the clustering of a corresponding set of points on a Grassmann manifold. Therefore, we call our approach the *Grassmannian Protein Clustering Method (GPCM)*. We evaluate the effectiveness of our method through experiments on the clustering of randomly selected proteins from the Protein Data Bank into four classes: alpha, beta, alpha/beta, alpha+beta (with multi-domain protein). The results show that GPCM outperforms the k-means clustering with Gauss Integrals Tuned, which is a state-of-the-art descriptor of protein structure.

Keywords: protein structure clustering, k-means, Mutual Subspace Method, Grassmann manifold, Gauss Integrals.

1 Introduction

Since there are numerous proteins whose functions are yet to be understood, accurately predicting protein structure and function is a main issue in structural bioinformatics. One important task in such computations is the clustering of 3D protein structures. In the clustering process, a distance metric is required to calculate the similarity between two proteins. The metric mostly used to measure the similarity between two protein structures is based on the root mean square deviation (RMSD) calculated from the coordinates of protein backbones. However RMSD raises problems in finding the best alignment and requires the superposition of two target proteins, which can be especially difficult when the shapes of the proteins are substantially different.

T. Shibuya et al. (Eds.): PRIB 2012, LNBI 7632, pp. 71–81, 2012.

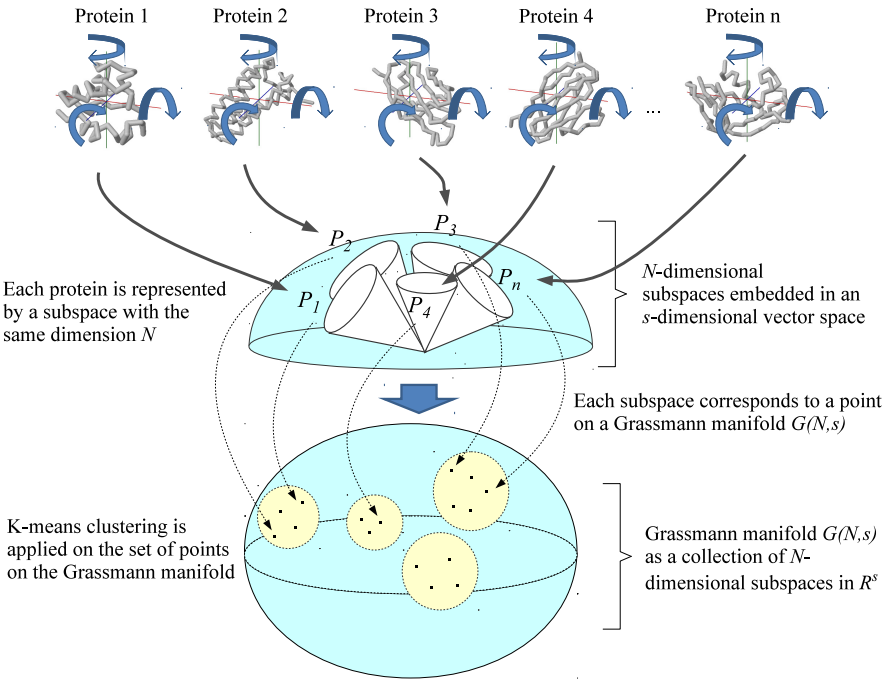

Fig. 1. The framework of the proposed method (GPCM)

There have been many attempts to establish an optimal alignment of protein structures based on RMSD [1][2][3], but there are few effective protein structure descriptors that overcome the limitations of RMSD. For example, when the 3D structure of a protein is represented by an oriented open curve in 3D space, a compact descriptor in the form of a 30-dimensional vector has been defined using two geometric measures, writhe and average crossing number [4]. This idea has been extended to a more robust descriptor, called Gauss Integrals Tuned (GIT) [5].

We propose a new method for clustering 3D protein structures in which the 3D structure of a protein is represented by a linear subspace (see Figure 1). Each subspace is generated using PCA from the set of synthesized multiple view images of the protein, as shown in Figure 2. The similarity of two protein structures is defined by the canonical angles between the corresponding subspaces. The advantage of this similarity measure [6] is that it does not rely on the precise alignment and geometry of the protein structures, so we can avoid the difficulties of the protein structure alignment. In this approach, we tackle the clustering problem of protein structure by considering the set of subspaces corresponding to the proteins. The clustering of subspaces with the same dimension is equivalent to the clustering of a corresponding set of points on a Grassmann manifold. Therefore, we call our approach the *Grassmannian Protein Clustering Method (GPCM)*.

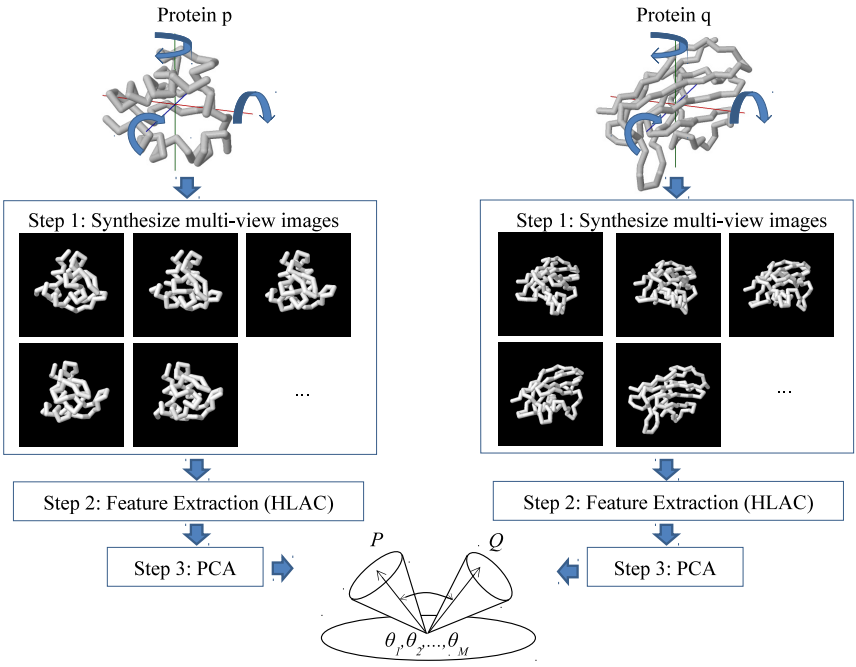

Fig. 2. Calculating the similarity of two protein structures based on canonical angles

The validity of the proposed method is demonstrated through experiments on the clustering of randomly selected proteins from the Protein Data Bank (PDB) into four protein fold classes: alpha (alpha-helices), beta (beta-sheets), alpha/beta (beta-alpha-beta motifs, mainly parallel beta-sheets), and alpha+beta (segregated alpha and beta regions, mainly anti parallel beta-sheets with some multi-domain proteins). The results show that our clustering method outperforms the conventional k-means clustering method with GIT [11], which is a state-of-the-art protein descriptor.

The organization of this paper is as follows. In Section 2 we provide a short description of the Gauss integrals descriptor. Then, in Section 3, we explain our approach which uses canonical angles to define protein structure similarity. Next, in Section 4 we outline the method of clustering subspaces using k-means on a Grassmann manifold. The experimental results are described and discussed in Section 5. Finally, in Section 6 we give some conclusions.

2 Protein Descriptor Based on Gauss Integrals

In the methodology of Gauss integrals as a protein descriptor, a protein backbone which is a trace of C_α carbon atoms is considered as an oriented open curve in space [4]. A series of 29 first, second, and third-order invariants, based on the generalized Gauss integrals for the writhe and average crossing number, are

computed over the curve. This set is considered as a vector $\in \mathcal{R}^{29}$ for interpreting the topology of 3-dimensional protein structure [5]. Including the number of residues, the final descriptor is a compact 30-dimensional feature vector of Gauss integrals. In order to make the Gauss integral based descriptor more robust to perturbations of protein structure, it has been extended to the Gauss Integrals Tuned (GIT) descriptor which uses a 31-dimensional vector [5]. In this paper, we focus on the k-means clustering method using Euclidean distances in the 31-dimensional GIT vector space.

3 Similarity Based on Canonical Angles

Canonical angles are also used as the similarity measure for image sets in the Mutual Subspace Method (MSM) [9][10]. The general procedure for using MSM to determine 3D protein structure similarity is as follows.

Let $\mathbf{x}_{i(i=1,...,n)}$ be an f-dimensional feature vector that belongs to protein p, where n is the number of samples. The basis vectors of an N-dimensional subspace $\mathcal{P}$ corresponding to protein p can be computed as the eigenvectors $[\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}]_N$ of the correlation matrix $\mathbf{A}$ [12]:

$$\mathbf{A} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \quad . \tag{1}$$

M canonical angles ($0 \leq \theta_1 \leq \ldots \leq \theta_M \leq \frac{\pi}{2}$) between an M-dimensional subspace $\mathcal{Q}$ and an N-dimensional subspace $\mathcal{P}$ ($M \leq N$) are defined as follows [7].

$$\cos\theta_i = \max_{\boldsymbol{u}_i \in \mathcal{Q}} \max_{\boldsymbol{v}_i \in \mathcal{P}} \boldsymbol{u}_i^{\mathrm{T}} \boldsymbol{v}_i \quad , \tag{2}$$

s.t. $\boldsymbol{u}_i^{\mathrm{T}} \boldsymbol{u}_i = \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{v}_i = 1, \boldsymbol{u}_i^{\mathrm{T}} \boldsymbol{u}_j = \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{v}_j = 0, i \neq j$.

In practice, we can obtain $cos^2\theta_i$ from the singular value of $\mathbf{P}^T\mathbf{Q}$, where $\mathbf{P} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N]$, $\mathbf{Q} = [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_M]$. Here $\boldsymbol{\phi}_i$ and $\boldsymbol{\psi}_i$ are the orthogonal basis vectors of the subspace $\mathcal{P}$ and $\mathcal{Q}$ respectively. The final similarity between two subspaces is given by

$$Sim = \frac{1}{M} \sum_{i=1}^{M} cos^2\theta_i \quad . \tag{3}$$

4 Algorithm of Clustering on a Grassmann Manifold

The standard k-means algorithm [14] attempts to partition a set of observation data $(\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_n)$ into k clusters $C_{i(i=1,...,k)}$ such that the sum of the distances among the data within each cluster is minimum:

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{\mathbf{d}_j \in C_i} ||\mathbf{d}_j - \mu_i||^2 \quad , \tag{4}$$

where μ_i is the mean of the data within cluster C_i.

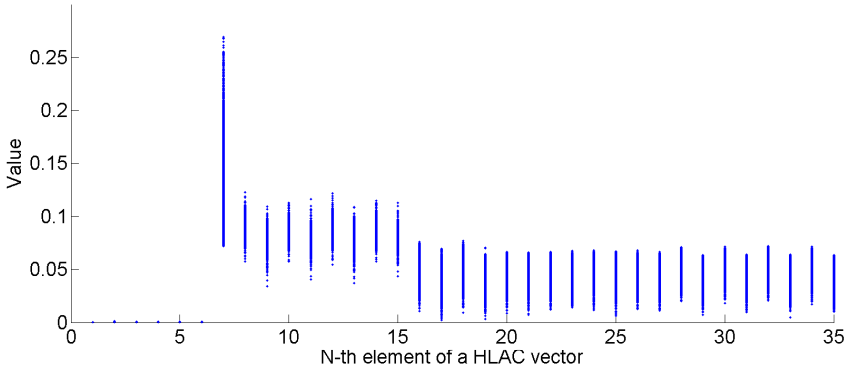

Fig. 3. Plot of 35-dimensional normalized HLAC feature vectors from various kinds of protein multiple view images. We removed the first 6 HLAC elements (the zeroth- and first-order correlations) which are close to zero.

In our problem, since each protein is represented by a linear subspace, we need to calculate the mean of multiple subspaces under the condition that the similarity of canonical angles should be regarded as a geodesic distance. Assume that an N-dimensional subspace $\mathcal{P}$ which lies in an s-dimensional vector space $\mathcal{R}^s$ corresponds to a point on the Grassmann manifold $G(N, s)$, and the subspace $\mathcal{P}$ is spanned by the columns of the $s \times N$ matrix $\mathbf{U}$. The mean of the points corresponding to subspaces on the Grassmann manifold can be obtained by using Algorithm 1[13], where $\mathrm{Glog}\,(\mathbf{X}, \mathbf{Y})$ can be calculated using Algorithm 2.

Algorithm 1. Computation of Karcher Mean on a Grassmann manifold [13]

1: Let $\mathbf{U}_{i(i=1,\ldots,n)} \in G(N, s)$ be the points on Grassmann manifold, and choose an error precision ϵ which is small enough (close to zero).
2: Initialize $\mu = \mathbf{U}_1$.
3: **repeat**
4: $\quad \delta = \frac{1}{N} \sum_{i1}^{N} \mathrm{Glog}\,(\mu, \mathbf{U}_i)$
5: $\quad$ Update $\mu = \mu \mathbf{V} \cos(\mathbf{S}) + \mathbf{U} \sin(\mathbf{S})$, where $\mathbf{USV}^T = \delta$
6: **until** $||\delta|| < \epsilon$

In summary, the flow of our proposed method is as follows.

Algorithm 2. $\mathrm{Glog}\,(\mathbf{X}, \mathbf{Y})$ [13]

1: $\mathbf{USV}^T = (\mathbf{I} - \mathbf{XX}^T)\mathbf{Y}(\mathbf{X}^T\mathbf{Y})^{-1}$
2: $\mathbf{\Theta} = \tan^{-1}(\mathbf{S})$
3: $\mathrm{Glog}\,(\mathbf{X}, \mathbf{Y}) = \mathbf{U\Theta V}^T$

Table 1. List of proteins used in the experiment. The first four characters are the PDB code and the last character indicates the chain ID of the protein.

Class	Protein List
Alpha (α)	1bbha,1hbga,1i3ea,1me5a,1qc7a,1s56a,1sr2a,2ccya,256ba,1tlha, 1jr5a,1c75a,1b7va,1k3ha,1k3ga,1enha,1hdpa,1ocpa,1b72a,1pufa
Beta (β)	1bioa,1d1ia,1exha,1ifca,1k1ja,1lcla,1mdca,1nsba,1rsub,1bwwa, 1b0wa,1b4ra,1ncia,1ncga,1op4a,1eeqa,1qaca,1ap2a,1cd0a,1pw3a
Alpha/Beta (α/β)	1aaza,1abaa,1g4ta,1hfra,1kofa,1mxia,1p2va,1rnha,1tcaa,1zona, 2foxa,3adka,3chya,2tpsa,1spqa,1v7za,1j2ta,1btaa,1h4xa,1h4za
Alpha+Beta ($\alpha+\beta$ and multi-domain proteins)	1apme,1atpe,8cata,1pfma,1r28a,1pu3a,178la,1hlea,1jtia,1as4a, 1qmna,1szqa,1qlpa,1opha,1hp7a,1bsca,1lxya,1ag2a,2baaa,3lzta

Step 1: For each protein synthesize multi-view backbone images of size 128×128 pixels by rotating the 3D model of the protein randomly around its viewing axes, using 3D molecular graphics software, Jmol [8].

Step 2: Extract a position-invariant feature vector, HLAC [15], from each multiple view image of the protein. Although the original HLAC is a 35-dimensional feature vector, which consists of several orders of local correlations, in this process we use a 29-dimensional HLAC starting from the second-order correlation. This is because the zeroth- and first-order elements of HLAC are almost zero, as shown in Figure 3. To deal with the diversity in the appearance and size of proteins, we empirically change the range for calculating local correlations from 1, 2, 3, 4, 5, 7, and 8 pixels, so that seven 29-dimensional HLACs are produced. Finally, we concatenate all the HLACs into a 203-dimensional HLAC feature vector.

Step 3: For each protein apply PCA to the set of 203-dimensional HLAC feature vectors to generate a subspace.

Step 4: Apply k-means clustering to the set of points on the Grassmann manifold corresponding to the set of subspaces.

5 Experiment

In the experiment, we randomly collected 80 proteins from the PDB site [16]. The test data are listed in Table 1. We applied our proposed clustering method, GPCM, and the conventional k-means clustering with GIT descriptor to this dataset. First we explain the details of the experimental conditions in Section 5.1. Then, the results of the clustering are discussed in Section 5.2. In Section 5.3, we discuss the results of an additional experiment.

5.1 Experimental Conditions

Since one protein may contain more than one chain, we first removed the unnecessary chain from each protein. Then, by following the flow of our framework from Step 1 to Step 3, as described in Section 4, we collected 3000 synthesized

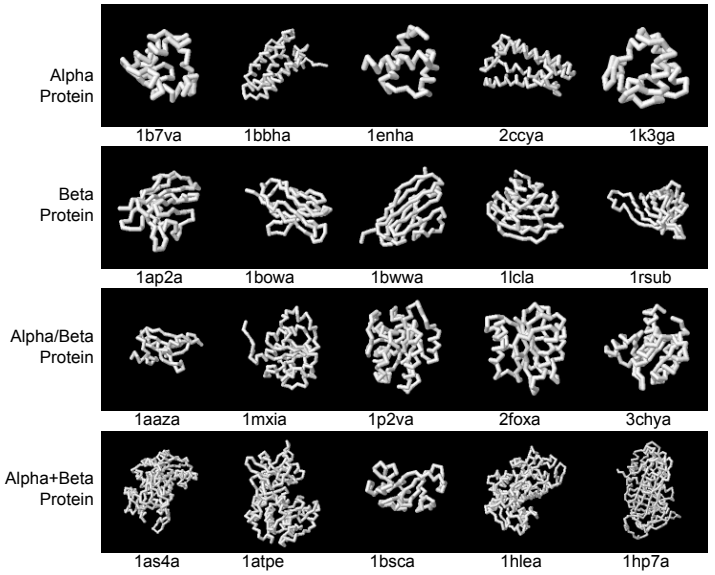

Fig. 4. Examples of the synthesized protein images used in the experiment

Table 2. Clustering results for the 80 proteins

Method	Measurement	Average (%)	Worst (%)	Best (%)
k-means with GIT	Accuracy	80.29	65.63	85
	Sensitivity	60.58	31.25	70
	Specificity	86.86	77.08	90
GPCM	Accuracy	**81.52**	**71.88**	**86.88**
	Sensitivity	**63.04**	**43.75**	**73.75**
	Specificity	**87.68**	**81.25**	**91.25**

protein images of size 128×128 pixels from each protein backbone visualization by using Jmol [8] which is included in the Matlab Bioinformatics Toolbox. Figure 4 shows some of the synthesized protein images. Next, we extracted HLAC vectors from these images to obtain 203-dimensional feature vectors. Finally, we constructed a subspace by applying PCA to each HLAC feature set. The dimension of the subspace was set to 4. Considering the randomness of the k-means clustering result, we repeated the clustering experiment 5000 times for both the proposed method and the k-means with GIT descriptor. The number of clusters was set to 4 ($k = 4$).

5.2 Clustering Results

Figure 5 shows box plots which summarize the accuracy of the clustering results from the experiment. Here, the clustering accuracy was defined as $(TP + TN)/(TP+TN+FP+FN)$, where TP is true positive, TN is true negative, FP

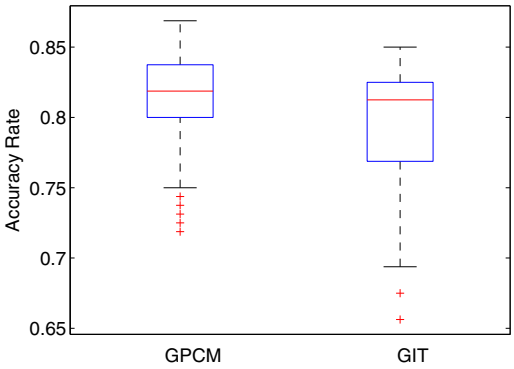

Fig. 5. Box plots of the 5000 experimental results using the proposed method and k-means with GIT descriptor

Table 3. The average true positive rate of each cluster from the 5000 repeated experiments. The columns indicate the clustering result. The rows indicate the ground truth label ($\alpha + \beta^*$ includes multi-domain proteins).

(a) GPCM

Class	α (%)	β (%)	α/β (%)	$\alpha+\beta^*$ (%)
α	71.15	0.13	22.03	6.69
β	0.3	71.98	7.77	19.95
α/β	22.75	8.19	51.84	17.22
$\alpha+\beta^*$	16.35	8.48	17.96	57.21

(b) k-means with GIT

Class	α (%)	β (%)	α/β (%)	$\alpha+\beta^*$ (%)
α	93.67	0	5.11	1.21
β	1.95	73.31	1.17	23.58
α/β	39.84	0.12	56.12	3.93
$\alpha+\beta^*$	28.6	51.12	1.05	19.22

is false positive, and FN is false negative. First, all possible combinations of the class labels for the clustering result were listed. Next, we computed the accuracy rate for each combination of class labels. Finally, the class label which produced the best accuracy rate was considered to be the correct label. The average, worst, and best clustering results of the GIT and the proposed method are shown in Table 2. The sensitivity (true positive rate) is defined as $TP/(TP + FN)$. The specificity (true negative rate) is defined as $TN/(FP + TN)$. We see that our proposed method is able to cluster the proteins more accurately than the conventional method. The proposed method could achieve up to 86.88% accuracy, 73.75% sensitivity, and 91.25% specificity. On the other hand, the conventional method achieved up to 85% accuracy, 70% sensitivity, and 90% specificity.

For further analysis, we examined the clustering results for each protein in both methods. Table 3 shows the average sensitivity (true positive rate) for each protein. These results show that the GIT descriptor is good at separating the alpha-helices and beta-sheets; however, it has serious difficulty clustering the overlapped structures of the fourth class which contains the alpha+beta proteins

Table 4. Clustering results for 400 proteins

Method	Measurement	Average (%)	Worst (%)	Best (%)
k-means with GIT	Accuracy	75	68	77
	Sensitivity	50.06	36	54
	Specificity	83.35	78.67	84.67
GPCM1	Accuracy	**75.08**	**71.13**	76.38
	Sensitivity	**50.16**	**42.25**	52.75
	Specificity	**83.39**	**80.75**	84.25
GPCM2	Accuracy	74.11	69	**77.88**
	Sensitivity	48.22	38	**55.75**
	Specificity	82.74	79.33	**85.25**

and the complicated multi-domain proteins. On the other hand, our proposed method has a more consistent performance across the categories. These results imply that there is room to improve the performance of our method by considering more effective features and tuning parameters, while the method with GIT may have some fundamental problems when dealing with overlapped and complicated protein structures. Moreover, the incapability of GIT to describe a protein which contains more than three consecutive missing carbon atoms is also a drawback of that method.

In terms of the computational speed, k-means with GIT is much faster than the proposed method. When using an Intel Xeon E5506 2.13Ghz and the Matlab statistical toolbox, the average execution time for the built-in k-means function with GIT descriptor was 0.0066s. On the other hand, our proposed method had an average execution time of 1.7s. However, it is worth noting that we have not optimized our Matlab implementation code to benefit from parallel processing, as we wrote our own implementation of k-means on the Grassmann manifold.

5.3 Additional Experiment

We conducted an additional experiment using 400 proteins. As in the previous experiment, we repeated the clustering 5000 times. However, in this experiment the fourth class of the protein does not contain multi-domain proteins (only alpha+beta proteins were used) to reduce the difficulty of classification. The experimental results for the clustering of the 400 proteins are shown in Table 4. GPCM1 used the same experimental parameters that were used in the experiment with 80 proteins. In GPCM2, the HLAC parameters were set to 2, 4, 6, and 8, and the subspace dimension was set to 5. Although the performance of the proposed method is quite similar to that of the conventional method in both cases, this experiment demonstrates that the performance of the proposed method can be improved by tuning the parameters of the subspace and having better feature extraction for the protein visualization images.

6 Conclusion and Future Work

In this paper, we have proposed a novel framework, called *Grassmannian Protein Clustering Method (GPCM)*, for solving the protein clustering problem. In GPCM, a 3D protein structure is represented by a linear subspace generated by applying PCA to the multiple-view of synthesized protein images. The similarity of two protein structures is defined by the canonical angles between the corresponding subspaces. The advantage of this approach is that it does not require precise alignment of the proteins. Since the protein is represented by a subspace, we regarded the protein clustering problem as a subspace clustering problem, and we applied the k-means algorithm for subspace clustering on a Grassmann manifold.

The experimental results demonstrated that the proposed method is superior to the conventional k-means with GIT approach, especially in identifying the overlapped structure of alpha+beta proteins which results a higher clustering accuracy. The GIT descriptor has a compact protein representation so the k-means computation is very fast. However, as shown by our experimental results, it may have a problem separating overlapped and complicated structures in which both alpha-helix and beta-sheet motifs exist.

Since this research is still in its early stages, we will conduct further experiments using more of protein data available from the PDB site. We will also consider different methods for extracting features from the protein visualization images and different classifiers, such as the nonlinear constrained subspace[17], with the aim of improving the performance of our method.

References

1. Holm, L., Sander, C.: DALI: a network tool for protein structure comparison. Trends Biochem. Sci. 20, 478–480 (1995)
2. Shindyalov, I., Bourne, P.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Engineering 11, 739–747 (1998)
3. Orengo, C.A., Taylor, W.R.: SSAP: Sequential structure alignment program for protein structure comparison. Methods in Enzymology 266, 617–635 (1996)
4. Røgen, P., Bohr, H.G.: A new family of global protein shape descriptors. Mathematical Biosciences 182(2), 167–181 (2003)
5. Røgen, P.: Evaluating protein structure descriptors and tuning Gauss Integrals based descriptors. Journal of Physics Condensed Matter 17, 1523–1538 (2005)
6. Suryanto, C.H., Jiang, S., Fukui, K.: Protein structures similarity based on multiview images generated from 3D molecular visualization. In: International Conf. on Pattern Recognition, ICPR 2012 (to appear 2012)
7. Chatelin, F.: Eigenvalues of matrices. John Wiley & Sons, Chichester (1993)
8. Jmol: an open-source Java viewer for chemical structures in 3D, `http://www.jmol.org/`
9. Yamaguchi, O., Fukui, K., Maeda, K.: Face recognition using temporal image sequence. In: International Conf. on Face and Gesture Recognition, pp. 318–323 (1998)

10. Fukui, K., Yamaguchi, O.: Face recognition using multi-viewpoint patterns for robot vision. In: 11th International Symposium of Robotics Research, pp. 192–201 (2003)
11. Harder, T., Borg, M., Boomsma, W., Røgen, P., Hamelryck, T.: Fast large-scale clustering of protein structures using Gauss Integrals. Journal of Bioinformatics, 510–515 (2012)
12. Oja, E.: Subspace Methods of Pattern Recognition. Research Studies Press, England (1983)
13. Begelfor, E., Werman, M.: Affine invariance revisited. In: Proceedings of International Conf. on Computer Vision and Pattern Recognition, pp. 2087–2094 (2006)
14. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Information Theory 28, 129–137 (1982)
15. Otsu, N., Kurita, T.: A new scheme for practical flexible and intelligent vision systems. In: Proc. of IAPR Workshop on CV, pp. 431–435 (1988)
16. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., et al.: The Protein Data Bank. Nucleic Acids Research 28, 235–242 (2000)
17. Fukui, K., Stenger, B., Yamaguchi, O.: A Framework for 3D Object Recognition Using the Kernel Constrained Mutual Subspace Method. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 315–324. Springer, Heidelberg (2006)