BIOINFORMATICS

Vol. 25 ISMB 2009, pages i276–i280
doi:10.1093/bioinformatics/btp219

A framework to refine particle clusters produced by EMAN

Liya Fan[1,2], Fa Zhang[1,*], Gongming Wang[1,2] and Zhiyong Liu[1,*]

[1]Institute of Computing Technology, Chinese Academy of Sciences and [2]Graduate University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

Motivation: EMAN is one of the most popular software packages for single particle reconstruction. But the particle clusters produced during its model refining stage are of low qualities. We attempt to refine the particle clusters by more accurately determining orientations of particles, and thereby achieving higher resolutions of consequent 3D structures.

Results: A particle reclustering framework (PRF) is introduced, which consists of three components. Each of them is responsible for one of the basic tasks of PRF: normalization, threshold determination and reclustering. Our implementation is also described and proved to meet the constraints proposed by PRF. Experiments revealed that our implementation improved resolutions of consequent structures for most cases, but only a little extra execution time was incurred. Therefore, it is practical to incorporate PRF in EMAN to improve qualities of generated 3D structures.

Availability and Implementation: Implementation of our algorithm is available upon request from the authors.

Contact: fanliya@ict.ac.cn; zf@ncic.ac.cn; zyliu@ict.ac.cn

1 INTRODUCTION

In many biological applications, it is required to determine 3D structures of protein macromolecules. So far, several technologies have been applied to deal with this problem, like X-ray crystallography, nuclear magnetic resonance (NMR) and electron microscopy single particle reconstruction. In recent years, electron microscopy single particle reconstruction has been gaining more and more popularities because it offers some advantages that are not available for other technologies. For example, it preserves the natural states of protein molecules, and suitable for protein molecules larger than 200 kDa (Ludtke *et al*., 1999).

Today, some software packages have been widely used in practice for single particle reconstruction, like EMAN (Ludtke *et al*., 1999), SPIDER (Frank *et al*., 1996), IMIRS (Liang *et al*., 2002), and so on. One of the major problems with them is that it is extremely time consuming to construct 3D structures by means of these tools (Scheres *et al*., 2007). Sometimes, each experiment may take several months.

One solution to this problem is to parallelize these programs, and conduct the computation by high-performance parallel computers. Some efforts have been made to address the problem in this way, like the parallel SPIDER program (Yang *et al*., 2007). The other solution is to improve the algorithm and accelerate the convergent process of the algorithms. As a result, fewer rounds of iteration are needed to get the desired resolution, or higher resolutions can be obtained within the same amount of time.

*To whom correspondence should be addressed.

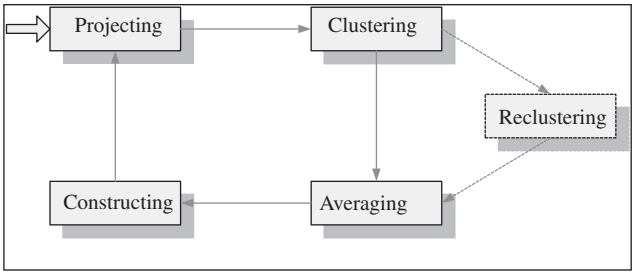


Fig. 1. Flow of the model refining stage of EMAN. Each round of the iteration consists of four steps: projecting, clustering, averaging and constructing. A new step called reclustering is added to the flow by us.

This study explores the way to accelerate the convergent process of EMAN. A particle reclustering framework (PRF) is introduced to improve the clusters produced by EMAN. By processing by PRF, the orientations of particles can be determined more accurately, so that the consequent 3D structures will have higher resolutions.

The rest of this article is organized as follows. Section 2 gives some general information about the algorithm of EMAN, and presents the problem. In Section 3, the framework is described and analyzed in details. Experimental results are shown in Section 4. Finally, we summarize and conclude in Section 5.

2 PROBLEM DESCRIPTION

For EMAN, three stages are involved in the process of single particle reconstruction (Ludtke *et al*., 1999). In the first stage, molecule particles are selected from micrographs. Second, an initial 3D model is generated. In the last stage, the initial model of the second step is refined through an iterative process. The third stage is the most time-consuming one; moreover, it directly decides the resolution of the final 3D structure.

The third stage can be further divided into four steps, as illustrated by Figure 1. First, projections of the initial model from various orientations are generated. Second, selected particles are grouped into a set of clusters with respect to the projections generated in the first step. Third, a class average is generated for each cluster. Finally, a new 3D model is constructed based on the class averages from the third step.

The basic operation of the third step is estimating the similarity between each projection and each particle. Specifically, a score s_{ij} is evaluated to reflect the similarity between the i-th projection and the j-th particle. The larger the score is, the closer the projection and particle are. For the following discussions, we suppose there are totally m projections and n particles. Thus, the j-th particle can be represented by an m-dimensional similarity vector

$$p_j = (s_{1j}, s_{2j}, \ldots, s_{mj})^T \quad 1 \le j \le n$$

The algorithm of EMAN finds the element, say s_{kj}, with the largest value in the vector. The cluster associated with the particle is then determined by the position of such element. In this example, it is k, and the corresponding cluster is c_k. Repeating this operation for each projection and particle, a set of m clusters $C=\{c_1,c_2, \ldots, c_m\}$ can be obtained. Each cluster corresponds to a projection, and contains the indices of particles that are most similar to the projection.

Intuitively, this method makes sense, because the position of the largest element represents the projection that is most similar to the particle. In reality, however, we found some problems. In some cases, one particle may be equally similar to more than one projection. The scores of the particle with respect to these projections are so close that it is difficult to tell which projection is more similar. A proof of the existence of such problem is that the same program compiled by different compilers produced different sets of clusters. Besides, for each similarity vector, only the largest score is utilized, and others are simply discarded, but they also provide important information about the particle.

In view of these problems, we introduced a framework called PRF to refine clusters provided by EMAN. After the clustering step of EMAN, each particle goes through an additional step called reclustering, if it satisfies some predefined conditions. Actual results show that the refined clusters determine the orientations of particles more accurately, and the consequent 3D structures have higher resolutions.

3 THE PARTICLE REFINING FRAMEWORK

In order to construct the framework, three key questions need to be answered. The first is how to normalize similarity vectors, so that the normalized vectors are more suitable to be processed by other methods, while preserving the information granted by similarity vectors. The second is how to decide which particles need to go through the additional reclustering step. The third is how to refine clusters by means of other methods. In the following sections, these questions are answered one after another.

3.1 Normalization

Similarity vectors have some properties that are not suitable to be processed by other algorithms. For example, different similarity vectors may have quite different value ranges, which make it unreasonable to compare different particles by directly comparing values in their similarity vectors. Normalization is aimed at solving these problems. It transforms similarity vectors to some other forms which can be processed more easily by other algorithms. This is achieved by a function $f: R^m \rightarrow R^m$.

Suppose $p=(s_1,s_2,\ldots,s_m)^T$ is the similarity vector for an arbitrary particle, and $q=(v_1,v_2,\ldots,v_m)^T$ is its normalized vector, namely $q=f(p)$, then function f should satisfy the following properties:

(i) For any $i_1,i_2 \in \{1, 2, \ldots, m\}$, if $s_{i1} \leq s_{i2}$, then $v_{i1} \leq v_{i2}$.
(ii) Suppose $v_{min}(j)=\min\{v_{ij}|1 \leq i \leq m\}$, and $v_{\max}(j)=\max\{v_{ij}|1 \leq i \leq m\}, 1 \leq j \leq n$, then $v_{\min}(1)=v_{min}(2)=\cdots=v_{\min}(n)$, and $v_{\max}(1)=v_{\max}(2)=\cdots=v_{\max}(n)$.

The first property means that the relative ranks of elements in the vector should be preserved after normalization. The second property makes sure that vectors of different particles have the same values ranges after normalization. In our implementation, the normalized vector $q=f(p)$ is carried out as follows:

$$s_{\min}=\min\{s_i|1 \leq i \leq m\} \quad (1)$$

$$s_{\max}=\max\{s_i|1 \leq i \leq m\} \quad (2)$$

$$t_i=\frac{s_i-s_{\min}}{s_{\max}-s_{\min}} \quad 1 \leq i \leq m \quad (3)$$

$$v_i=\frac{e^{t_i}-1}{e-1} \quad 1 \leq i \leq m \quad (4)$$

It can be verified that this implementation satisfies the two properties. First, property (i) is satisfied because formulas (3) and (4) are both monotonous increasing functions. It can be easily proved that $\min\{t_i|1 \leq i \leq m\}=0$, and $\max\{t_i|1 \leq i \leq m\}=1$, so we can get $\min\{v_i|\ 1 \leq i \leq m\} = 0$ and $\max\{v_i|\ 1 \leq i \leq m\} = 1$. Because p is an arbitrary similarity vector, this proof can be applied to any particles. Therefore, $v_{\min}(1)=v_{\min}(2)=\cdots=v_{\min}(n) = 0$ and $v_{\max}(1) = v_{\max}(2) = \cdots = v_{\max}(n) = 1$ are true, and property (ii) is verified.

3.2 Threshold determination

This section deals with the problem of choosing particles for reclustering. Intuitively, the more uncertain the cluster of a particle is, the more likely the particle will participate in reclustering. We define the clustering certainty of a particle as the difference between values of the largest and second largest elements in its normalized vector. Therefore, the clustering certainty can be used as a criterion to decide whether a particle needs reclustering. For a particle with a large clustering certainty, the most similar projection can be easily identified, so reclustering is not necessary for it. On the other hand, for a particle with a small cluster certainty, the difference between the largest two elements in the normalized vector is trivial, and the most similar projection is unclear, so reclustering is needed to help identify the correct cluster.

Once the clustering certainty is known for each particle, a threshold can be determined, and the decision of choosing particles for reclustering can be made. If the clustering certainty of a particle is greater than the threshold, then it skips the reclustering step, with its cluster remaining unchanged; otherwise, it participates in reclustering.

The threshold can be determined in one of the two ways. First, the threshold can be a fixed constant. Second, the threshold can take the value so that a fixed percentage of particles participate in reclustering. In our implementation, both methods were used, and corresponding results will be given in Section 4.

3.3 Reclustering

This section describes the method to determine the cluster associated with each particle that participates in reclustering. Here, we borrowed the idea of the K-Means algorithm (Willett, 1980), one of the most popular algorithms for clustering. The idea of K-Means is simple and easy to implement, but a major problem is how to properly select initial centroids, so as to determine initial clusters, because this will greatly influence qualities of the final clusters. In many applications of K-Means, initial centroids are chosen randomly, as a result, qualities of the consequent clusters are usually low and unstable.

However, that is not a problem for the case here. We take the clusters produced by the clustering step of EMAN ($c_1, c_2, \ldots, c_m$) as initial clusters. This is reasonable because for most cases, clusters produced by EMAN are close enough to the final 'corrected' clusters. Suppose n normalized vectors $q_1, q_2, \ldots, q_n$ are available, obtaining initial centroids is simple and straightforward. Each centroid can also be represented as an m-dimensional vector:

$$s_i = \frac{1}{|c_i|} \sum_{j \in c_i} q_j \quad i = 1, 2, \ldots, m \tag{5}$$

The distance between the j-th particle and the i-th centroid can be characterized by norm of the vector $q_j - s_i$. Besides, similarity scores should also be incorporated in the distance function because they also reflect the distances between particles and centroids. This is accomplished by giving a group of weights $w_1, w_2, \ldots, w_m$, to each particle, and incorporating them in the distance function as:

$$d(q, s_i) = w_i \| q - s_i \| \quad i = 1, 2, \ldots, m \tag{6}$$

where q is an arbitrary normalized vector. The first term of (6) represents the clustering decision made by similarity scores, and the second term reflects the preference of the K-Means algorithm. Suppose $q = (v_1, v_2, \ldots, v_m)^T$, the weights should have the property that if $v_{i1} \leq v_{i2}$, then $w_{i1} \geq w_{i2}$, for any $1 \leq i_1, i_2 \leq m$. In our implementation, we set

$$w_i = \begin{cases} \frac{1}{v_i} & v_i \neq 0 \\ +\infty & v_i = 0 \end{cases} \tag{7}$$

There are many definitions for norms of vectors. In our implementation, we chose the infinity norm, which is defined as:

$$x = (x_1, x_2, \ldots, x_n)^T \tag{8}$$

$$\|x\|_\infty = \lim_{p \to \infty} \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} = \max\{|x_i| \mid 1 \leq i \leq m\} \tag{9}$$

By evaluating distances to all centroids, each particle's associated cluster can be determined. Specifically, for a particle with the normalized vector q, if its associated cluster is c_k, then the following equality must hold.

$$d(q, s_k) = \min\{d(q, s_i) \mid 1 \leq i \leq m\} \tag{10}$$

It can be noticed that all values of the normalized vectors are made use of during the reclustering process, which overcomes the shortcoming of the algorithm used by EMAN.

3.4 Summary of the framework

So far, every detail of the framework has been described. Our implementation has also been given, and proved to meet the constraints proposed by each component of the framework. The framework can be summarized as follows:

```
PRF(P={p1,p2 ... pn}, C={c1,c2 ... cm})
1   for j = 1 to n do
2       qj = normalize (pj)
3       ccj = clustering_certainty(qj)
4   endfor
5   threshold = get_threshold(cc1, cc2 ... ccn)
6   for i = 1 to m do
7       si = (1/|ci|) Σ_{j∈ci} qj
8   endfor
9   for j = 1 to n do
10      if ccj ≤ threshold then
11          for i = 1 to m do
12              d(qj, si) = wij ||qj − si||
13          endfor
14          d(qj, sk) = min{d(qj, si) | 1 ≤ i ≤ m}
15          Find the cluster cl, so that j∈cl
16          cl ← cl − {j}
17          ck ← ck ∪ {j}
18      endif
19  endfor
```

Inputs of PRF are the similarity vectors for all particles, as well as the set of clusters provided by EMAN. The loop of lines 1–4 calculates the normalized vector and clustering certainty for each particle. For our implementation, it takes $O(m)$ time to calculate a normalized vector, and $O(m)$ to calculate the clustering certainty, so the total time of the loop of lines 1–4 is $O(mn)$. The threshold is evaluated in line 5, of which the time complexity is $O(n)$, if a fixed percentage of particles participate in reclustering, or $O(1)$ if the threshold is chosen to be a constant. The loop of lines 6-9 calculates a set of centroids. Any centroid s_i can be determined in time $O(m|c_i|)$, by observing

$$m \sum_{i=1}^{m} |c_i| = mn \tag{11}$$

the total time of the loop is $O(mn)$.

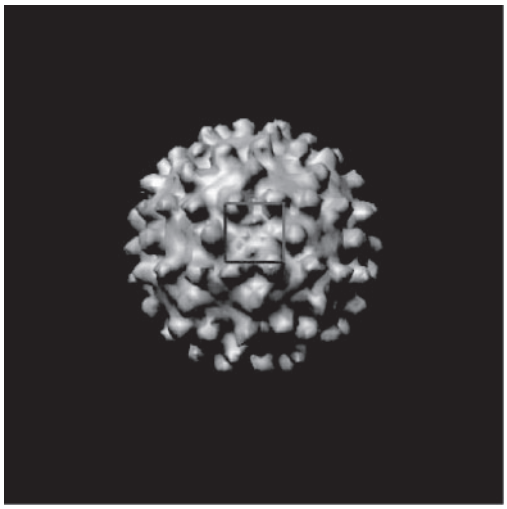
The task of reclustering is performed by the loop of lines 9–19. It first compares the clustering certainties with the threshold, only particles with cluster certainties smaller than or equal to the threshold go through the rest of the iteration. The inner loop of lines 11–13 calculates the distances to all centroids. In our implementation, a distance can be calculated in time $O(m)$, so the total time for line 12 cannot exceed $O(nm^2)$. Line 14 finds the smallest distance to all centroids, which also decides the new cluster of the particle. This operation can be finished in time $O(m)$. Lines 15–17 get the old cluster of the particle, and update contents of both the old and new clusters. Each of these operations can be finished in time $O(1)$, if implemented properly. Therefore, the loop of lines 9–19 takes $O(nm^2)$ time in total.

Based on analysis of three parts of the framework, we can conclude that the time complexity of our implementation is $O(nm^2)$.

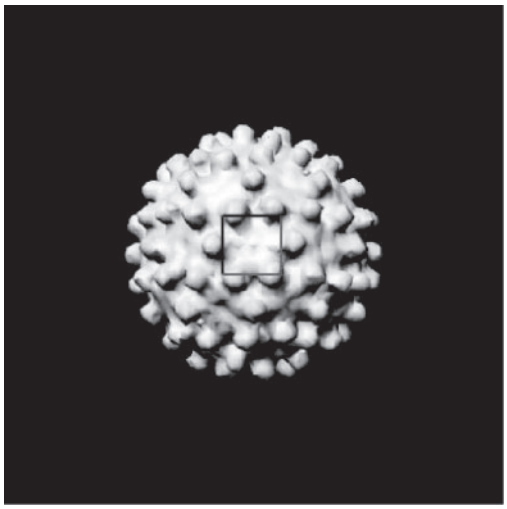
4 EXPERIMENTAL RESULTS

PRF has been applied to actual protein single particle reconstruction experiments. This section presents results of these experiments. In order to identify the benefits gained by PRF, experimental results of the original EMAN program are also given.

In our experiments 10 datasets were used, five of which were initial 3D models of the Hepatitis B Virus, and the other five were initial 3D models of the Chaperonin β subunit from the Thermoacidophilic Archaeon. They were generated by other experiments of single particle reconstruction. Specifically, for each of the two macromolecules, EMAN was applied to one initial model, and five rounds of iteration were executed, which produced five new 3D models. These new 3D models acted as our initial 3D models in



(**a**)



(**b**)

Fig. 2. The 3D structures of HBV1 produced by different algorithms. (**a**) The structure produced by the original EMAN program, and (**b**) the structure processed by PRF. It can be noticed from the parts within red boxes that structure processed by PRF is smoother, with less noise.

the following experiments. In this study, they are denoted by HBV1, HBV2, …, HBV5, and Beta1, Beta2, …, Beta5.

For experiments of the Hepatitis B Virus, 4239 selected particles were used, which were grouped into 379 clusters, and for experiments of the Chaperonin β subunit, 2334 particles were grouped into 85 clusters. The two strategies for evaluating the threshold were both adopted in our experiments. For Hepatitis B Virus, a fixed percentage (10%) of particles participated in reclustering, and for Chaperonin β subunit, a fixed constant threshold (0.005) was used.

Figure 2 gives an example of comparison between 3D structures produced by different algorithms. Figure 2a is the 3D structure of HBV1 constructed by the original EMAN program, and Figure 2b is the structure of HBV1 processed by PRF. By comparing the parts within red boxes, it can be noticed that the structure processed by PRF is smoother, with less noise.

Table 1. Resolutions of structures of Hepatitis B Virus

Dataset	Orig-EMAN	PRF
HBV1	10.9221	10.6633
HBV2	10.0515	9.75645
HBV3	9.18231	9.20543
HBV4	9.23309	8.932203
HBV5	9.29348	9.29836

Table 2. Resolutions of structures of Chaperonin β subunit

DATASET	ORIG-EMAN	PRF
Beta1	33.5746	32.9683
Beta2	32.2496	31.5523
Beta3	31.1206	31.0678
Beta4	32.6046	32.5191
Beta5	31.3382	31.1551

For all datasets, we compared resolutions of their consequent structures produced by different algorithms. Resolutions of these structures were estimated by means of Fourier shell correlation (Penczek, 1998), as shown in Tables 1 and 2. The column of Orig-EMAN corresponds to the results of the original EMAN program, and the column of PRF corresponds to the program with PRF.

From Tables 1 and 2, we can see that for 8 of the 10 datasets, PRF processed structures had higher resolutions. The greatest increase was about 0.3 Å for Hepatitis B Virus, and 0.7 Å for Chaperonin β subunit. The reason might be that PRF corrected some particles' associated clusters, so their orientations were determined more accurately, which lead to higher qualities of consequent structures.

For one dataset (HBV3), PRF processed structure had a resolution slightly lower than the structure produced by the original EMAN program. For the remaining dataset (HBV5), resolutions of the PRF processed structure and the structure generated by Orig-EMAN were almost the same. This may be due to the fact that reclustering may mistakenly decide clusters of particles in some cases. For example, when too many particles' clusters are incorrect, the centroids calculated by PRF are incorrect either. Or if a particle is equally close to more than one projection according to the distance function of the framework. In that case, PRF may randomly select a cluster, which gives rise to an incorrect cluster.

We implemented PRF in C++ programming language on Linux platform, and Table 3 displays the execution time for each dataset. For all datasets, the longest time was <40 s, which is trivial compared with other steps of EMAN. This implies that PRF can be used in practice to improve 3D structures produced by EMAN, without incurring too much extra execution time.

5 CONCLUSIONS

In this study, a PRF is introduced to refine clusters produced by EMAN, with the purpose of determining orientations of particles more accurately and achieving higher resolutions of consequent 3D

Table 3. Execution time of our implementation of PRF

Dataset	Time (s)	Dataset	Time (s)
HBV1	31.029	Beta1	10.623
HBV2	32.562	Beta2	10.334
HBV3	32.027	Beta3	10.358
HBV4	31.629	Beta4	10.521
HBV5	32.194	Beta5	10.532

structures. It has three components, each one dealing with a basic problem. First, a function is required to convert similarity vectors to normalized vectors. The function must satisfy two conditions described in Section 3.1. Second, a criterion is needed to choose particles for reclustering. This is accomplished by obtaining a threshold and comparing the clustering certainty of each particle with it. The last component determines the associated clusters with particles by means of a new method. The clustering decisions made by EMAN must also be taken into account by the new method.

Our implementation of PRF is provided, and corresponding results reveals that for most cases, this implementation is capable of improving resolutions of consequent 3D structures. Moreover, this implementation does not incorporate too much extra execution time. All these suggest that PRF can be applied in practice to improve resolutions of 3D structures produced by EMAN, and hence accelerate the convergent process of EMAN.

ACKNOWLEDGEMENTS

We would like to thank Fei Sun for providing the experimental datasets, and Steven Ludtke and Wen Jiang for critical reading of the manuscript.

Funding: National Natural Science Foundation for China (90612019, 60752001, 60736012 and 60503060); Chinese Academy of Sciences knowledge innovation key project (KGCX1-YW-13).

Conflict of Interest: none declared.

REFERENCES

Frank,J. *et al.* (1996) SPIDER and WEB: processing and visualization of images in 3D electron microscopy and related fields. *J. Struct. Biol.*, **116**, 190–199.

Liang,Y. *et al.* (2002) IMIRS: a high-resolution 3D reconstruction package integrated with a relational image database. *J. Struct. Biol.*, **137,** 292–304.

Ludtke,S.J. *et al.* (1999) EMAN: semiautomated software for high-resolution single-particle reconstructions. *J. Struct. Biol.*, **128**, 82–97.

Penczek,P. (1998) Measures of resolution using Fourier shell correlation. *J. Mol. Biol.*, **280**, 115–116.

Scheres,S.H.W. *et al.* (2007) Disentangling conformational states of macromolecules in 3D-EM through likelihood optimization. *Nat. Methods*, **4**, 27–29.

Willett,P. (1980) Document clustering using an inverted file approach. *J. Inform. Sci.*, **2**, 223–231.

Yang,C. *et al.* (2007) The parallelization of SPIDER on distributed-memory computers using MPI. *J. Struct. Biol.*, **157**, 240–249.