Journal of Structural Biology xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Journal of Structural Biology

journal homepage: www.elsevier.com/locate/yjsbi

Gaussian-input Gaussian mixture model for representing density maps and atomic models

Takeshi Kawabata

Institute for Protein Research, Osaka University, 3-2 Yamadaoka, Suita, Osaka 565-0871, Japan

ARTICLE INFO

Keywords:
Electron microscopy
Fitting
Superposition
Atomic model
Gaussian mixture model
Expectation maximization algorithm

ABSTRACT

A new Gaussian mixture model (GMM) has been developed for better representations of both atomic models and electron microscopy 3D density maps. The standard GMM algorithm employs an EM algorithm to determine the parameters. It accepted a set of 3D points with weights, corresponding to voxel or atomic centers. Although the standard algorithm worked reasonably well; however, it had three problems. First, it ignored the size (voxel width or atomic radius) of the input, and thus it could lead to a GMM with a smaller spread than the input. Second, the algorithm had a singularity problem, as it sometimes stopped the iterative procedure due to a Gaussian function with almost zero variance. Third, a map with a large number of voxels required a long computation time for conversion to a GMM. To solve these problems, we have introduced a Gaussian-input GMM algorithm, which considers the input atoms or voxels as a set of Gaussian functions. The standard EM algorithm of GMM was extended to optimize the new GMM. The new GMM has identical radius of gyration to the input, and does not suddenly stop due to the singularity problem. For fast computation, we have introduced a down-sampled Gaussian functions (DSG) by merging neighboring voxels into an anisotropic Gaussian function. It provides a GMM with thousands of Gaussian functions in a short computation time. We also have introduced a DSG-input GMM: the Gaussian-input GMM with the DSG as the input. This new algorithm is much faster than the standard algorithm.

1. Introduction

Single-particle electron microscopy generates high-resolution 3D density maps for large biomolecular complexes. Fitting an atomic model or another 3D map onto the map is important to extract biological insights. However, because a 3D density map has a large number of grid points with densities, the 3D fitting often requires a long computation time. To reduce the computation time, several coarse-grained or shape approximation methods have been proposed. The most popular approach is the placement of a set of 3D points (3D point model). The vector quantization method is widely used for approximating the molecular shapes of both atomic models and density maps (Wriggers et al., 1998). The K-means method is also popular, and generates a set of 3D points (Lasker et al., 2009). De-Alarcón et al. proposed a kernel density estimation algorithm for vector quantization (De-Alarcón et al., 2002). They regarded a 3D point as an isotropic Gaussian function with the same variance (homoscedastic function), and employed a maximum likelihood method (EM algorithm) to estimate the parameters. Jonić and Sorzano employed a set of homoscedastic isotropic Gaussian functions with weights (Jonić and Sorzano, 2016), and proposed a heuristic algorithm to optimize not only the centers and weights, but also the number of Gaussian functions. Many 3D fitting programs, such as SITUS (Wriggers, 2012) and γ-TEMPy (Pandurangan et al., 2015), employ the 3D point model, because it can solve a continuous optimization problem of rotations and translations as a combinatorial optimization problem of matching points. The 3D point model (or a set of isotropic Gaussian functions with weights) is also suitable for an elastic network model to study dynamics and deform a density map (Tama et al., 2002; Ming et al., 2002a,b). Jin et al. employed a set of isotropic Gaussian functions with weights to perform the elastic 3D-to-2D alignment (Jin et al., 2014). Joubert and Habeck proposed a method for reconstructing the initial 3D density map from 2D images, using isotropic homoscedastic Gaussian functions with weights (Joubert and Habeck, 2015). Jonić et al. employed an isotropic Gaussian function to reduce the noise in high-resolution density maps (Jonić et al., 2016).

A Gaussian mixture model (GMM) has been utilized to approximate a 3D density map and an atomic model (Kawabata, 2008). GMM is a probability distribution function with a weighted sum of multiple Gaussian functions, and is widely used for statistical modeling and classification (McLachlan and Peel, 2000; Bishop, 2006; Gupta and Chen, 2011). We have developed the program *gmfit* for fitting density maps and atomic models using GMM (Kawabata, 2008). We also have

E-mail address: kawabata@protein.osaka-u.ac.jp.

https://doi.org/10.1016/j.jsb.2018.03.002
Received 25 May 2017; Received in revised form 2 February 2018; Accepted 3 March 2018

developed a WEB server for fitting pairs of density maps and atomic models (Suzuki et al., 2016). The program IMP (Integrative Modeling Platform) also employs GMM for fitting (Lasker et al., 2010; Russel et al., 2012; Robinson et al., 2015). GMM has several advantages over the 3D point model. First, because a component of the mixture model is an anisotropic Gaussian function, it can approximate a given shape using a smaller number of components, as compared to isotropic 3D points. Even one 3D Gaussian function has a shape property in three eigen values of its covariance matrix, whereas one 3D point or isotropic Gaussian function has no shape property. Second, GMM has an efficient and solid algorithm for the fitting, the expectation–maximization (EM) algorithm, which maximizes a likelihood function (Dempster et al., 1977). Third, an overlap integral of two GMMs can be analytically obtained, and the integral is differentiable with respect to the translation and rotation of the GMM (Kawabata, 2008). This property is beneficial for finding the best fitting configuration using a derivative-based technique.

However, the standard EM algorithm for GMM has several problems. First, it accepts points as its input without considering their size. A voxel of a 3D density map and an atom of an atomic model are handled simply as a center point, and the width of the voxel and the spherical radius of the atom are ignored. Second, the EM algorithm has the so-called "singularity problem", as it sometimes stops its iterative procedure when one of the Gaussian functions has a very small spread and zero-division occurs (Bishop, 2006; Gupta and Chen, 2011). When a 3D Gaussian function almost corresponds to the input points on the plane, the determinant of its covariance matrix is close to zero, and then the value of the Gaussian function becomes unstable. In the 3D space, three or fewer points are always on the plane. The GMM singularity problem has been discussed in several textbooks (Bishop, 2006; Gupta and Chen, 2011). These books mentioned that the EM algorithm sometimes fails by approaching singularities of the log-likelihood function, especially when the number of observations is small relative to the number of Gaussian functions. The use of suitable ad hoc heuristics, such as reinitializing means randomly, or employing a MAP EM (maximum *a posterori* EM) method by introducing a prior distribution, was recommended. The singularity problem is not observed frequently, however, when hundreds of density maps must to be converted to GMMs, often some of them will have singularity problems. We observed many singularities, while developing the server for searching and comparing EMDB data (Suzuki et al., 2016). Since a singularity stops the repetition of the EM algorithm before convergence, it often provides a GMM that is less similar to the original map. Third, the EM algorithm requires a long computation time for converting a density map with a large number of grid points. For example, only 5 s are required to convert a 64^3 voxel map to 10 Gaussian functions, whereas six hours are required for converting a 512^3 voxel map to 10 Gaussian functions.

To solve these problems, we now introduce a Gaussian-input GMM algorithm with inputs that are a set of Gaussian functions corresponding to voxels or atoms. The variance of the input Gaussian function are determined by those of a voxel cube or an atomic sphere. The likelihood function can be optimized by an extension of the standard EM algorithm of the GMM. This new algorithm is expected to generate a GMM with the identical variance to the input voxels and atoms, and to be free from the singularity problem. To enhance the computation speed to convert a large 3D density map into a GMM, we introduce a down-sampled Gaussian functions (DSG) by merging neighboring voxels into an anisotropic Gaussian function. We also introduce a down-sampled Gaussian-input GMM, which uses the DSG as the input. This new down-sampling algorithm is expected to generate a better model than that obtained from a simple down-sampled map.

2. Methods

2.1. Outline of EM algorithm for Gaussian mixture model

The goal of this study is to estimate the parameters of a Gaussian mixture model to approximate the shape of a 3D density map or an atomic model. We want to reproduce the observed data (3D density map or atomic model), using the probability of the model $P(\boldsymbol{r}|\boldsymbol{\theta})$ defined as follows:

$$P(\boldsymbol{r}|\theta) = \sum_{k}^{K} w_k \phi_k(\boldsymbol{r}), \tag{1}$$

where $\boldsymbol{r}$ is a 3D vector, $\phi_k(\boldsymbol{r})$ is the k-th 3D Gaussian function, w_k is the weight for $\phi_k(\boldsymbol{r})$, K is the number of Gaussian functions, and $\boldsymbol{\theta}$ indicates a set of parameters. The k-th Gaussian function is defined as follows:

$$\begin{aligned}\phi_k(\boldsymbol{r}) &= \phi(\boldsymbol{r}|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) \\ &= \frac{1}{(2\pi)^{3/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{r}-\boldsymbol{\mu}_k)\right],\end{aligned} \tag{2}$$

where $\boldsymbol{\mu}_k$ is a 3D vector of the center position, and $\boldsymbol{\Sigma}_k$ is a 3×3 covariance matrix. If we have to clearly indicate the parameters for the function, we can utilize $\phi(\boldsymbol{r}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, instead of $\phi_k(\boldsymbol{r})$. The value of the weight w_k ranges from 0 to 1, and their sum should be 1:

$$\sum_{k}^{K} w_k = 1. \tag{3}$$

The parameter set $\boldsymbol{\theta}$ for the GMM is defined as $\boldsymbol{\theta} = \{\{w_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\}\}$. Input data can be classified into three types. These three types of GMM for both the density map and atomic model are schematically summarized in Fig. 1. i) Point-input (P-input; Fig. 1B and F). The input is a set of 3D positions $\{\boldsymbol{r}_i\}$. ii) Weighted point-input (WP-input; Fig. 1C and G). The input is a set of 3D positions $\{\boldsymbol{r}_i\}$ with weights $\{\rho_i\}$. iii) Gaussian-input (G-input; Fig. 1D and H). The input is a set of 3D Gaussian functions $\{\phi_i(\boldsymbol{r})\}$ with weights$\{\rho_i\}$. The point-input GMM is the most standard GMM. Many textbooks have described its algorithm, and the expectation-maximization (EM) algorithm can efficiently optimize the likelihood of the model (McLachlan and Peel, 2000; Bishop, 2006; Gupta and Chen, 2011). The weighted point-input GMM is also solved by a simple extension of the EM algorithm of the point-input GMM (Kawabata, 2008), and its details are described in the accompanying Supplementary Information. The EM algorithms for the point-input and the weighted point-input GMMs are summarized in Fig. 2. The EM algorithm iterates the E (expectation)-step and the M (maximization)-step (Dempster et al., 1977). The E-step estimates the posterior probability that the i-th sample was generated by the k-th Gaussian function. The posterior probabilities $h_{ki}^{(t)}$ for the point-input and the weighted point-input GMM are identical, where t is the number of iterations. The M-step updates the parameters of GMM, w_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$. The equations for the updates are also summarized in Fig. 2. Note that the equation of posterior probability (denoted as $h_i(\mathbf{r}_t)$) in Kawabata (2008) was incorrect, and this mistake was fixed in Fig. 2.

In the next subsection, we will explain the EM algorithm of the Gaussian-input GMM in detail. For readers unfamiliar with the standard Gaussian mixture model, the Supplementary Information includes the algorithm for the weighted point-input GMM and the basic mathematics. For readers uninterested in the mathematical details of the algorithm, the equations in Fig. 2 are the minimum requirements to develop Gaussian-input GMM computer programs.

2.2. Likelihood function for Gaussian-input GMM

Next, we explain the likelihood function for the Gaussian-input GMM. The input of the Gaussian-input GMM is a set of Gaussian functions $\{\phi_i(\boldsymbol{r})\}$ with weights $\{\rho_i\}$. The number of functions (the length of $\{\phi_i(\boldsymbol{r})\}$ and $\{\rho_i\}$) is N. Each weight ρ_i is assumed to be non-negative.