Journal of Molecular Graphics and Modelling 38 (2012) 411–418

Contents lists available at SciVerse ScienceDirect

Journal of Molecular Graphics and Modelling

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

Triangulated manifold meshing method preserving molecular surface topology

Minxin Chen[a,**], Bin Tu[b], Benzhuo Lu[b,*]

[a] Center for System Biology, Department of Mathematics, Soochow University, Suzhou 215006, China
[b] State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics, Academy of Mathematics and Systems Science, National Center for Mathematics and Interdisciplinary Sciences, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Accepted 29 September 2012
Available online 10 October 2012

Keywords:
TMSmesh
Triangulated manifold
Molecular surface meshing
Gaussian surface
Numerical simulation

ABSTRACT

Generation of manifold mesh is an urgent issue in mathematical simulations of biomolecule using boundary element methods (BEM) or finite element method (FEM). Defects, such as not closed mesh, intersection of elements and missing of small structures, exist in surface meshes generated by most of the current meshing method. Usually the molecular surface meshes produced by existing methods need to be revised carefully by third party software to ensure the surface represents a continuous manifold before being used in a BEM and FEM calculations. Based on the trace technique proposed in our previous work [1], in this paper, we present an improved meshing method to avoid intersections and preserve the topology of the molecular Gaussian surface. The new method divides the whole Gaussian surface into single valued pieces along each of x, y, z directions by tracing the extreme points along the fold curves on the surface. Numerical test results show that the surface meshes produced by the new method are manifolds and preserve surface topologies. The result surface mesh can also be directly used in surface conforming volume mesh generation for FEM type simulation.

1. Introduction

A variety of problems in computational structural biology require molecular surface meshes, such as in surface property visualization of a biomolecule, docking and certain implicit solvent modelling. In implicit-solvent simulations, high-quality surface meshes are critical for BEM and FEM type approaches to produce convergent and reasonable results. Recent developments of fast algorithms in this field make molecular surface mesh generation for large biomolecules with good qualities become a more urgent issue than the mesh generations only for the purposes of visualization or some structural geometry analysis [2]. Especially in mathematical modelling, good quality is usually needed to be guaranteed. Evincing the properties of a continuous manifold is one desirable feature of a high-quality mesh. A surface is a manifold if each point has a neighborhood homeomorphic to a disk in the real plane. Meshing a manifold surface should also produce a manifold mesh. Here, a manifold mesh means that the surface formed by all the elements of the mesh is also a manifold. Some examples of common mesh structures that are incompatible with the manifold definition are shown in Fig. 1.

Various definitions of molecular surface exist, including the van der Waals surface, solvent accessible surface (SAS) [3], solvent excluded surface (SES) [4], molecular skin surface [5,6] and Gaussian surface, etc.

For SAS and SES computations, the representative methods are as follows. Connolly's method is the pioneer work on calculating the molecular surface and SAS analytically [7,8]. In 1995, a popular program, GRASP, for visualizing molecular surfaces was presented [9]. An algorithm named SMART for representing SAS as a smooth triangulated manifold was proposed by Zahar [10]. Sanner et al. presented a tool based on α shapes [11], named MSMS, for meshing the SES [12]. In 1997, Vorobjev et al. proposed SIMS, a method of calculating a smooth invariant molecular dot surface, in which an exact method for removing self-intersecting parts and smoothing the singular regions of the SES was presented [13]. Ryu et al. proposed a method based on Beta-shapes, which is a generalization of alpha shapes [14]. Can et al. proposed LSMS to generate the SES on grid points using level-set methods [15].

For Skin surface, Chavent et al. presented MetaMtal to visualize the molecular skin surface using ray-casting method [16], and Cheng et al. used restricted union of balls to generate mesh for molecular skin surface [17].

For the Gaussian surface, existing techniques for triangulating implicit surface can be used to mesh the Gaussian surface. These methods are divided into two main categories: spatial partition and continuation methods. The well known marching cubes method [18] and dual contouring method [19] are examples of the spatial

* Corresponding author. Tel.: +86 10 62626492; fax: +86 10 62542285.
** Co-corresponding author.
E-mail addresses: chenmx@gmail.com (M. Chen), tubin@lsec.cc.ac.cn (B. Tu), bzlu@lsec.cc.ac.cn (B. Lu).

1093-3263/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jmgm.2012.09.006

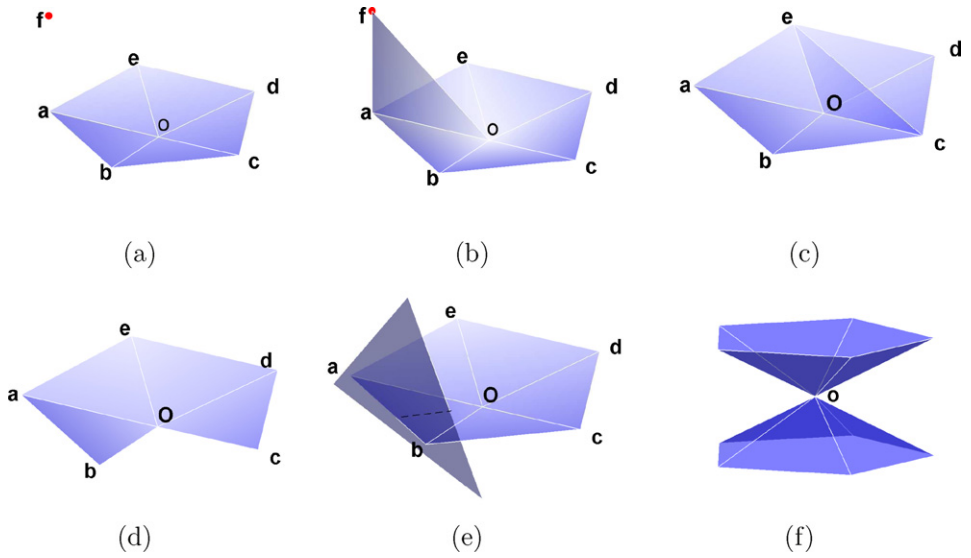

Fig. 1. Some examples of non-manifold meshes. In (a), the red point f is an isolated vertex in the mesh. In (b), the edge $a-o$ is shared by three triangles in the mesh. In (c), triangle $a-o-c$ is an abundant face in the mesh. In (d), triangle $b-o-c$ is missed in the mesh. In (e), intersecting triangle pairs occur in the mesh. In (f), the point o is shared by two cones in the mesh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

partition methods. These methods divide three dimensional space into cells and assume an underlying continuous function to define the surface, typically derived from a spatial density. Polygonal faces approximating the surface are generated at appropriate cells by examining the signature of the signs of the function at the corners of each cell. It is assumed that the underlying function is approximately linear in each cell. In 2006, Zhang et al. used a modified dual contouring method to generate mesh for biomolecular structures [20]. The continuation methods [21–23] are of another category. These methods mesh the implicit surface by growing current polygonization's border through adding new vertices. The quality of mesh triangles is well controlled in these meshing processes, but techniques for avoiding overlaps, filling the gap between adjacent branches and selecting proper initial triangles are required.

Recently, we have proposed a method, named TMSmesh, for triangular meshing of the Gaussian surface [1]. The trace technique which is a generalization of predictor-corrector technique is used in TMSmesh to connect sampled surface points. In TMSmesh, there are no problems of overlapping, gap filling, and selecting initial seeds that need to be considered compared with traditional continuation methods, because the Gaussian surface is polygonized by connecting presampled surface points. TMSmesh is capable of finishing meshing Gaussian surface for biomolecules consisting of more than one million atoms with grid space of 1 Å within 30 min on a typical 2010 PC, and the mesh quality is shown to be adequate for BEM simulations of bimolecular electrostatics.

But one issue is that defects commonly exist in the meshes generated by existing methods, including TMSmesh. Here we classify the defects into two main categories. The first is that the molecular surface mesh is not a manifold. This kind of defects leads to problems in BEM and FEM type simulations of biomolecules. For example, the computational results produced by BEM are not reasonable around these defective places and the volume mesh generation conforming to the surface always fails due to these defects. The second category is that the topology of a surface mesh is not exactly consistent with the original surface topology, for example some small structures like holes and tunnels smaller than grid spaces are missed. The second kind of defect often causes the first kind of defect (see (a) and (b) in Fig. 2). The topology defect is also a problem for adaptive meshing strategy for the molecular surface, because the topology of a mesh at the fine level may be found to be not consistent with that at the coarse level (see (a) and (c) in Fig. 2). To avoid these defects of surface meshes, an improved version of TMSmesh is proposed in this paper by introducing a technique of dividing the whole Gaussian surface into single valued patches along each of x, y, z directions through tracing the extreme points along the fold curves. Each single value patch is easily decomposed into non-intersecting triangles with standard polygon triangulation methods as shown in [24]. Moreover, tracing the additional extreme points facilitates finding of surface structures such as holes and tunnels smaller than the grid spacing.

This paper is organized as follows. We present our improved method for meshing the Gaussian surface in Section 2. Some examples, analyses and applications are presented in the Section 3. The final section, Conclusion, gives some concluding remarks.

2. Method

Before introducing the new method, we briefly review main points of our previous methods in TMSmesh to help connect to our improved method. The technical details of TMSmesh can be referred to [1].

2.1. Previous algorithms in TMSmesh

TMSmesh is designed for meshing the molecular Gaussian surface. The Gaussian surface is defined as a level set of the summation of the Gaussian kernel functions as follows:

$$\{\vec{x} \in \mathbb{R}^3, \phi(\vec{x}) = t_0\}, \tag{1}$$

where

$$\phi(\vec{x}) = \sum_{i=1}^{N} e^{-d(||\vec{x}-\vec{c}_i||^2/r_i - 1)}, \tag{2}$$

the parameter d is positive and controls the decay speed of the kernel functions, $\vec{c}_i$ and r_i are the location and radius of atom i. In this work, the value of d and t_0 are set as 0.5 and 1, respectively.

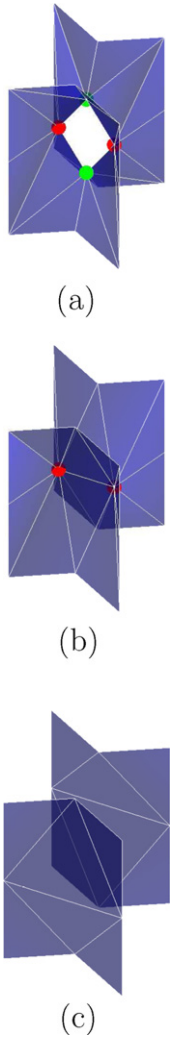

Fig. 2. Three possibly generated meshes of a same surface with a hole are illustrated. (a) Shows a triangulated manifold in which the hole is correctly presented. (b) Shows a non-manifold mesh of the same surface in which the neighborhood of each red vertex is not homeomorphic to a disk. The hole is collapsed in (b) due to that the green vertices in (a) are missed in (b). (c) Shows a mesh missing the hole of the surface. The green and red vertices in (a) are missed, due to that grid space is coarser than that of the mesh in (a). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Compared with other definitions of molecular surface, Gaussian surface is smooth and the surface properties, such as the Gaussian and mean curvatures, can be easily computed [25]. The VDW surface, SAS and SES can be approximated well by the Gaussian surface with proper parameter selection [25,26]. The Gaussian surface has been widely used in many problems in computational biology, such as docking problem [27], molecular shape comparison [28], calculating SAS area [29] and the generalized Born models [30].

TMSmesh contains two main steps. The first step is to compute the intersecting points between the molecular Gaussian surface and the lines parallel to x-axis. In this step, the molecule is placed in a three-dimensional orthogonal grid. The upper and lower bounds in each box of the implicit function $\phi(x)$ is estimated in order to rule out the boxes having no surface points. In the remaining cubes, the intersect points between the surface and the lines parallel to x-axis are found through root finding algorithms. In the second step, the sampled surface points are connected through two algorithms to form loops, because the whole closed manifold surface can be decomposed into a collection of pieces enclosed by loops on the surface. The first is the trace step which connects two adjacent topological connected surface points on the lines parallel to x-axis. In the trace step the predictor-corrector method is used to generate the next corrected surface point from the current one, and the topology connection is confirmed by checking the continuity between the corrected and the current points. If the continuity is not fulfilled, we restart the predictor-corrector from the current point with a smaller step size. During this trace process, some extreme point along α direction ($\alpha \in \{x, y, z\}$) in the surface, i.e. the point $\vec{x}_0$ satisfying $\frac{\partial\phi(\vec{x})}{\partial\alpha}|_{\vec{x}=\vec{x}_0} = 0$ and $\phi(\vec{x}_0) = t_0$, is also found by checking the sign change of $\frac{\partial\phi(\vec{x})}{\partial\alpha}$ on the traced path. These extreme points form an extreme point list. The trace processes need to be done in both $x-y$ and $y-z$ planes to complete the construction of surface patches. The second algorithm gives the sequence of the trace steps on xy and xz planes to ensure the connected surface points on lines parallel to x-axis forming loops on the surface. With these two algorithms, all the sampled surface points are connected and formed close loops on surface. These loops are treated as polygons on the surface in TMSmesh. When the distances between the lines parallel to x-axis, i.e. the grid spaces, are small enough, the patches enclosed by the loops are mostly likely single valued pieces. But patches enclosed by the loops may contain holes and tunnels, and in the step of triangulating the patch, these holes or tunnels may be missed. In addition, intersections may occur. In these cases, we found the patches are always non-single valued patches along at least one of x, y, z directions. The non-single valued patches cause difficulties to generating triangulated manifold and preserving the topology of the surface. Even the grid space is very small the above problems can not be avoided completely. So we proposed the following improved method to dissect each patch enclosed by loop into single valued pieces in x, y, z directions further.

2.2. An improved method

Compare with the old version of TMSmesh, in the new method, the tracing technique is not only used to connect the sampled intersecting points between the surface and parallel lines but also to connect extreme points along the fold curves formed by extreme points on the surface. The fold curves for x, y, z directions are defined as follows.

$$\{\vec{x} \in \mathbb{R}^3, \phi(\vec{x}) = t_0, \frac{\partial\phi(\vec{x})}{\partial\alpha} = 0\}, \alpha \in \{x, y, z\}. \tag{3}$$

The Gaussian surface defined in Eq. (1) is folded at these fold curves for x, y, z directions. Cutting the Gaussian surface along these fold curves ensures the resulted pieces are single valued on x, y, z directions. Then, these pieces can be easily triangulated through standard triangulation algorithms. Moreover, tracing along the fold curves also help find the missed small surface structures, such as tunnels and holes smaller than grid space, since these small structures also fold at these curves. For instance, in Fig. 3, the red curves may indicate existence of a half hole and the hole will be found by this technique. We will introduce the algorithm as follows (see Algorithm 1 and Fig. 3). Without loss of generality, suppose we have the initial extreme point $\vec{x}_0^* = (x_0^*, y_0^*, z_1)$ in the extreme point list found by previous trace algorithm in old version of TMSmesh [1] on the plane:

$$\begin{cases} z = z_1 \\ y_0 < y < y1 \end{cases} \tag{4}$$

satisfying

$$\begin{cases} \phi(\vec{x}_0^*) = t_0 \\ \dfrac{\partial\phi(\vec{x}_0^*)}{\partial x} = 0 \end{cases}, \tag{5}$$

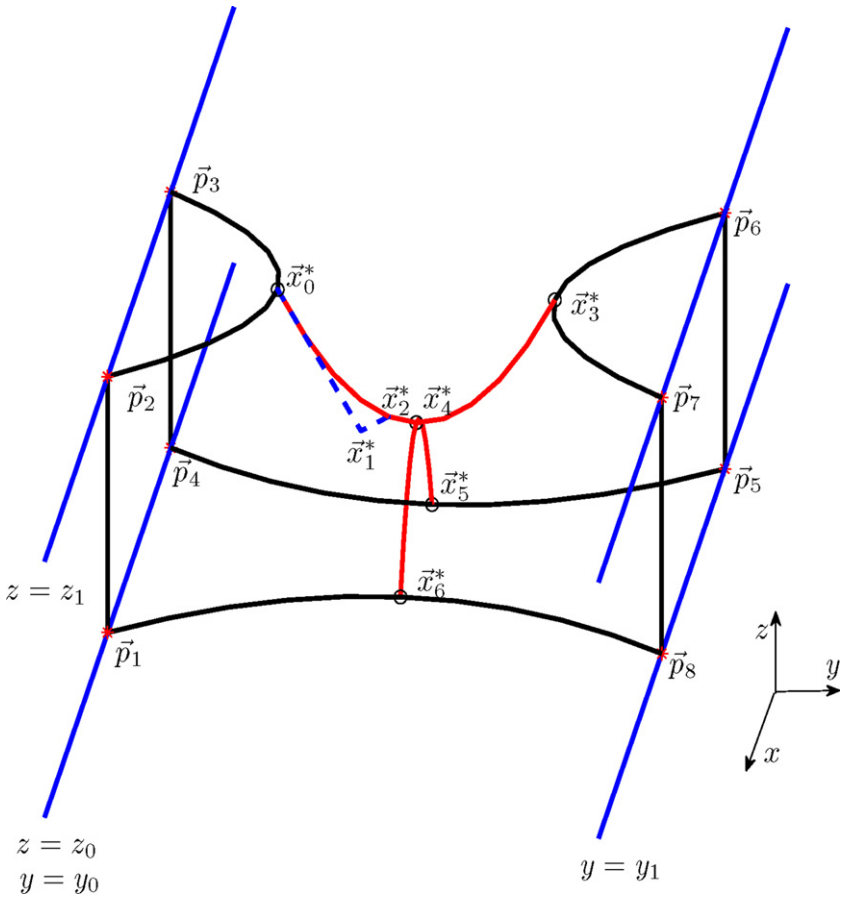

Fig. 3. An example of connecting surface extreme points along the fold curves use Algorithm 1. The red curves are fold curves. The black curves form a close loop on the surface. The surface patch enclosed by the black loop is not single valued along x, y directions and it is divided into four single valued pieces along x, y directions by the fold curves. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

where y_0, y_1, z_0, z_1 are y-coordinates and z-coordinates of four adjacent lines parallel to x-axis. Algorithm 1 is used to find the connected extreme point of $\vec{x}_0^*$ that is the intersection point between the fold curve

$$\{\vec{x} \in \mathbb{R}^3, \phi(\vec{x}) = t_0, \frac{\partial \phi(\vec{x})}{\partial x} = 0\}. \tag{6}$$

and one of the following four planes:

$$\begin{cases} y = y_0 \\ z_0 < z < z_1 \end{cases}, \begin{cases} y = y_1 \\ z_0 < z < z_1 \end{cases}, \begin{cases} z = z_0 \\ y_0 < y < y_1 \end{cases},$$

$$\begin{cases} z = z_1 \\ y_0 < y < y_1 \end{cases}. \tag{7}$$

Algorithm 1. Tracing step for connecting the extreme points

Input: Initial step size h_1 and initial extreme point $\vec{x}_0^* = (x_0^*, y_0^*, z_1)$ on the surface. The y-coordinates and z-coordinates of four adjacent lines, y_0, y_1 and z_0, z_1. User defined small positive value ε and the bound for the cosine value δ ($0 < \delta < 1$).

Step 1, initialize $h_2 = h_1$ and $\vec{x}_0^* = (x_0^*, y_0^*, z_1)$.

Step 2, let $\vec{x}_1^* = \vec{x}_0^* + h_2 sig \cdot \vec{T}$, where $\vec{T}$ is the tangent direction of the fold curve (Eq. (6)) at $\vec{x}_0^*$, and $\vec{T} = (\phi_{zx}(\vec{x}_0^*)\phi_y(\vec{x}_0^*) - \phi_{yx}(\vec{x}_0^*)\phi_z(\vec{x}_0^*), -\phi_{xx}(\vec{x}_0^*)\phi_z(\vec{x}_0^*), -\phi_{xx}(\vec{x}_0^*)\phi_y(\vec{x}_0^*))$ (here ϕ_x means partial derivative of ϕ with respect to x), sig is the sign of $-\phi_{xx}(\vec{x}_0^*)\phi_y(\vec{x}_0^*)$ to make the direction of $sig \cdot \vec{T}$ toward plane $z = z_0$.

Step 3, use Newton iterations to correct $\vec{x}_1^*$ to the fold curve, i.e., find $\vec{x}_2^*$ satisfying Eq. (6) through taking $\vec{x}_1^*$ as the initial point of Newton iterations of solving the nonlinear equation $\phi_x^2(\vec{x}) + (\phi(\vec{x}) - t_0)^2 = 0$.

Step 4, if $|\phi_\alpha(\vec{x}_2^*)| < \varepsilon$, $\vec{x}_2^*$ is an extreme point along α direction,$\alpha \in \{y, z\}$, add it to the extreme point list.

Step 5, if $\cos(\nabla\phi(\vec{x}_0^*), \nabla\phi(\vec{x}_2^*)) < \delta$ (condition $\star_1$)
or ($\phi_\alpha(\vec{x}_0^*)\phi_\alpha(\vec{x}_2^*) < 0$ and $\min(|\phi_\alpha(\vec{x}_0^*)|, |\phi_\alpha(\vec{x}_2^*)|) > \varepsilon, \alpha \in \{y, z\}$)
(condition $\star_2$), let $h_2 = h_2/2$ and go to step 2.

Step 6, if the line segment between $\vec{x}_0^*$ and $\vec{x}_2^*$ crosses one of the planes in Eq. (7), interpolate them to get the intersection point on the plane $\vec{x}_3^*$. Check if $\vec{x}_3^*$ is in the extreme point list, if not, add it to the extreme point list, stop.

Step 7, let $\vec{x}_0^* = \vec{x}_2^*$ and go to step 1.

Output: The first connected surface point $\vec{x}_3^*$ on plane described by one of Eq. (7), and the extreme point(s) along y, z directions, if exist.

In step 2 of Algorithm 1, $\vec{x}_1^*$ is the predicted surface point from $\vec{x}_0^*$ along the tangent direction of fold curve on the surface (Eq. (6)) with step size h_2. Step 3 is to correct $\vec{x}_1^*$ back to the fold curve (Eq. (6)). Step 4 is to check whether $\vec{x}_2^*$ is an extreme point along the y, z directions. In step 5, condition ($\star_1$) is used to determine if the step size h_2 is acceptable through checking whether the angle between the normal directions at $\vec{x}_0^*$ and $\vec{x}_2^*$ is small enough, otherwise restart the prediction and correction from $\vec{x}_0^*$ with a smaller step size; δ is a user-specified bound for cosine value of the angle. If the condition is not sufficient to decide the continuity in some cases, other conditions can be added, such as continuity of higher order derivatives. In our practical computation, condition ($\star_1$) is sufficient. In the case that an extreme point along the y or z direction exists between $\vec{x}_0^*$ and $\vec{x}_2^*$, condition ($\star_2$) is used to detect it. In step 6, if line segment between $\vec{x}_0^*$ and $\vec{x}_2^*$ crosses one of the planes in Eq. (7), then the intersect point $\vec{x}_3^*$ is the final traced surface extreme point we want to connect to. In step 7, $\vec{x}_0^*$ is replaced by $\vec{x}_2^*$ and step

Table 1
Description of molecules in the PQR benchmark.

Molecule (name or PDB code)	Number of atoms	Description
FAS2	906	fasciculin2, a peptide inhibitor of AChE
AChE monomer	8280	mouse acetylcholinesterase monomer
AChE tetramer	36638	the structure of AChE tetramer, taken from [33]
30S ribosome	88431	30S ribosome, the PDB code is 1FJF
70S ribosome	165337	obtained from 70S_ribosome3.7A_model140.pdb.gz on http://rna.ucsc.edu/rnacenter/ribosome_downloads.html
3K1Q	203135	PDB code, a backbone model of an aquareovirus virion
2X9XX	510727	a complex structure of the 70S ribosome bound to release factor 2 and a substrate analog, which has 4 split PDB entries: 2X9R, 2X9S, 2X9T, and 2X9U [34]
1K4R	1082160	PDB code, the envelope protein of the dengue virus [35]

1 is repeated to trace the next connected point on the fold curve (Eq. (6)). Algorithm 1 is repeated until all points in the extreme points list are connected. These extreme points include those found through our previous method and the additional extreme points on fold curves found by the Algorithm 1.

Fig. 3 shows an example of connecting extreme points $\vec{x}_0^*$ and $\vec{x}_3^*$ that locate in the plane (Eq. (4)) and the fold curve (Eq. (6)). In Fig. 3, $\vec{p}_i, i = 1, \cdots, 8$ are surface points on the four adjacent blue lines parallel to x-axis. $\vec{p}_i, i = 1, \cdots, 8$ and extreme points $\vec{x}_0^*, \vec{x}_3^*, \vec{x}_5^*, \vec{x}_6^*$ are vertices of the close loop connected by previous algorithms in TMSmesh [1]. The surface bounded by this close loop is not single valued either along x or y direction, since the surface folds at the red curves. The red curve connecting $\vec{x}_0^*, \vec{x}_4^*$ and $\vec{x}_3^*$ is a fold curve on the surface satisfying $\phi_x(\vec{x}) = 0$. The red curve connecting $\vec{x}_5^*, \vec{x}_4^*$ and $\vec{x}_6^*$ is a fold curve on the surface satisfying $\phi_y(\vec{x}) = 0$. These two fold curves divide the surface bounded by the black close loop into four single valued polygons along x, y-directions: $\vec{p}_1 - \vec{p}_2 - \vec{x}_0^* - \vec{x}_4^* - \vec{x}_6^*$, $\vec{p}_7 - \vec{p}_8 - \vec{x}_6^* - \vec{x}_4^* - \vec{x}_3^*, \vec{p}_4 - \vec{p}_3 - \vec{x}_0^* - \vec{x}_4^* - \vec{x}_5^*, \vec{p}_5 - \vec{p}_6 - \vec{x}_3^* - \vec{x}_4^* - \vec{x}_5^*$. The original patch bounded by the black close loop is triangulated poorly, with flat elements that deviate dramatically from the underlying isosurface, and that with the aid of the fold curves the geometry of the patch is better defined. $\vec{x}_1^*$ is the predicted point along the tangent direction of extreme point curve from $\vec{x}_0^*, \vec{x}_2^*$ is the corrected surface extreme point from $\vec{x}_1^*$. $\vec{x}_4^*$ is the surface extreme point with $\phi_x(\vec{x}_4^*) = 0$ and $\phi_y(\vec{x}_4^*) = 0$, it can be found by the condition $(\star_2)$ in Algorithm 1 during the trace process along the fold curves.

Algorithm 1 indicates that the connected extreme points on the fold curves can be located through trace step from initial extreme point, and the location of the initial extreme point is the only one condition to initialize the trace process. For this reason, the surface fold structure smaller than grid space will not be missed unless no initial extreme point on the fold structure is detected in the previous method. Fig. 4 shows some examples of tunnel structures smaller than grid space are found by the current method on the surface of 3K1Q molecule (see Table 1).

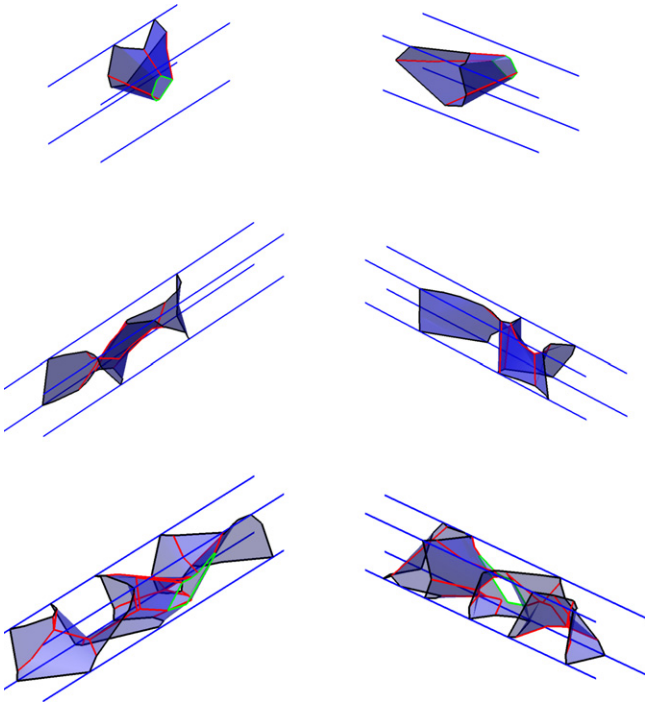

Fig. 4. Examples of tunnel structures smaller than grid space found by the new version of TMSmesh. Each row is a same local surface structure seen from different angles of view. The black curves are close loops formed by connected intersection points of the surface and blue lines parallel to x-axis. The red ones are fold curves traced by Algorithm 1. These red curves divide the surface into single-valued pieces along x, y, z directions. The green curves are close loops formed by extreme points connected by the trace steps on the corresponding plane. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

3. Results

Because MSMS is a typical and widely used tool for surface meshing in molecular modelling, in this section, the qualities of meshes generated by the old and the new versions of TMSmesh are compared to those of MSMS. It is worth making a note here that MSMS operates on SES and TMSmesh on a Gaussian surface. To make a reasonable comparison with MSMS, grid spaces of 1.0 and 0.7 Å are chosen for TMSmesh to approximate the surface vertex densities 1/Å^2 and 2/Å^2 used in MSMS mesh generation, respectively. The probe radius in MSMS is set to be 1.2Å. All computations run on Dell Precision T7500 with Intel® Xeon® CPU 3.3 GHz and 48 GB memory under 64 bit Linux system. Firstly, we check whether the meshes produced by TMSmesh and MSMS are triangulated manifolds. Following are three necessary conditions for a mesh to be a manifold mesh.

a) The first is that each edge should be shared and only be shared by two faces (a face denotes an element patch) of the mesh.

b) The second is that each vertex should have and only have one neighborhood node loop. Neighborhood node of a vertex is a node connecting the vertex through an edge. For instance, in Fig. 1 (a), points a, b, c, d, e are neighborhood nodes of vertex o, and loop $a-b-c-d-e$ is the neighborhood loop of vertex o. Clearly, in Fig. 1(c) and (f) are two examples in which the vertex o has more than one neighborhood loops.

c) The third is that the mesh has no intersecting face pairs.

For a triangular mesh, each triangle is a face. Table 2 shows the number of non-manifold defects and number of intersecting triangle pairs in the meshes produced by MSMS and the old and new versions of TMSmesh. The meshing softwares are run on the molecular PQR files (PDB +atomic charges and radii information). For tests of the mesh tool in this work, we use our previous PQR benchmark (see Table 1) that can be found and is downloadable at our web page http://lsec.cc.ac.cn/lubz/Meshing.html.

It is shown in Table 2 that TMSmesh is a robust tool for meshing the Gaussian surface for large biomolecules. The meshes produced by the new version of TMSmesh all satisfy the above three necessary conditions for a manifold mesh, while the meshes of large

Table 2
Number of non-manifold errors in meshes produced by TMSmesh and MSMS.[a]

Molecule	Number of atoms	Number of non-manifold defects[b]			Number of intersecting triangle pairs		
		TMSmesh		MSMS	TMSmesh		MSMS
		Old	New		Old	New	
FAS2	906	0	0	0	0	0	0
		0	0	0	0	0	0
AChE monomer	8280	0	0	0	0	0	52
		0	0	0	0	0	27
AChE tetramer	36638	0	0	20	35	0	534
		0	0	7	10	0	1203
30S ribosome	88431	19	0	0	65	0	1820
		2	0	2	0	0	3858
70S ribosome	165337	30	0	25	406	0	4560
		17	0	Fail	281	0	–
3K1Q	203135	8	0	0	105	0	1721
		7	0	Fail	149	0	–
2X9XX	510727	44	0	Fail	852	0	–
		16	0	Fail	433	0	–
1K4R	1082160	6	0	Fail	463	0	–
		54	0	Fail	788	0	–

[a] The data in the first and second row for each molecule are corresponding to densities 1 vertex/Å^2 and 2 vertex/Å^2, respectively.
[b] Number of vertices whose neighborhood does not satisfy aforementioned necessary conditions a), b) for a manifold mesh (see the text in Section 3).

molecules produced by the old version of TMSmesh and MSMS do not.

Table 3 shows the CPU timings of the old and the new versions of TMSmesh and MSMS for the entire PQR benchmark set, and the number of vertices of each mesh is also listed. It is shown that the numbers of vertices produced by the new version of TMSmesh are more than those produced by the old version at the same grid spacing, because the new version of TMSmesh samples the fold curves while the old version does not. The discrepancies between the numbers of verities of TMSmesh mesh and MSMS mesh are due to different definitions of molecular surface and different meshing methods used in the two programs. Meanwhile the CPU time required to run the new version of TMSmesh is much more than that to run the old version of TMSmesh. The reasons are as follows. First, each step of tracing along the fold curves costs much more than tracing on the surface. In Algorithm 1, both the first and second order directives need to be computed at each predictor-corrector step, while the trace step on the surface only need compute the first order directives. And in step 3 of Algorithm 1, the required number of Newton iterations is much more than that of Newton iterations in tracing step on surface, as both $\phi_x(\vec{x})$ and $\phi(\vec{x}) - t_0$ are required to be sufficiently small simultaneously, while $\phi(\vec{x}) - t_0$ is the only term required to be zero in the Newton iterations in tracing step on the surface. The trace step along fold curves (Algorithm 1) is employed intensively in the new version of TMSmesh. Therefore, a much expensive CPU time is required additionally by the new version of TMSmesh in the trace steps along the fold curves. Secondly, due to inclusion of additional nodes traced on the fold curves, the mesh has more vertices than that generated by the old version TMSmesh, hence needs more CPU time in the polygonization and triangulation steps. TMSmesh is expected to be speeded up by further tuning the code, and by using an adaptive box structure, parallel computing, and more sophisticated predictor-corrector algorithm.

To verify that the meshes produced by the new version of TMSmesh preserve the topology of a molecular surface, for two medium sized molecules, AChE monomer and AChE tetramer, the Euler characteristics are calculated for meshes produced by the new and old versions of TMSmesh and MSMS with different vertex densities from 1/Å^2 to 10/Å^2. Fig. 5 indicates that the Euler characteristic of the meshes produced by the new version of TMSmesh converges more smoothly than those of meshes produced by MSMS and the old version of TMSmesh when the vertex density increases.

Table 3
CPU time comparison and number of vertices of the meshes generated by TMSmesh and MSMS.[a]

Molecule	Number of atoms	Number of vertices			CPU time(s)		
		TMSmesh		MSMS	TMSmesh		MSMS
		Old	New		Old	New	
FAS2	906	2484	5170	3599	1.3	8	0.1
		4082	8309	6766	1.8	10	0.1
AChE monomer	8280	11704	24556	38689	12	65	0.6
		22538	39289	55343	19	75	0.8
AChE tetramer	36638	46178	95433	141810	40	280	5.9
		87654	152035	204132	62	325	7.9
30S ribosome	88431	133957	274297	372557	92	901	16.2
		256072	439020	555342	151	1400	19.1
70S ribosome	165337	336497	698055	901122	180	1523	46.2
		640720	1111399	Fail	283	1702	–
3K1Q	203135	246502	509390	753022	226	1800	51.5
		470743	812774	Fail	359	2160	–
2X9XX	510727	762461	1585434	Fail	577	6012	–
		1450535	2521233	Fail	970	7203	–
1K4R	1082160	1600336	3325975	Fail	1260	9120	–
		3048432	5298234	Fail	2080	16132	–

[a] The data in the first and second row for each molecule are corresponding to densities 1 vertex/Å^2 and 2 vertex/Å^2, respectively.

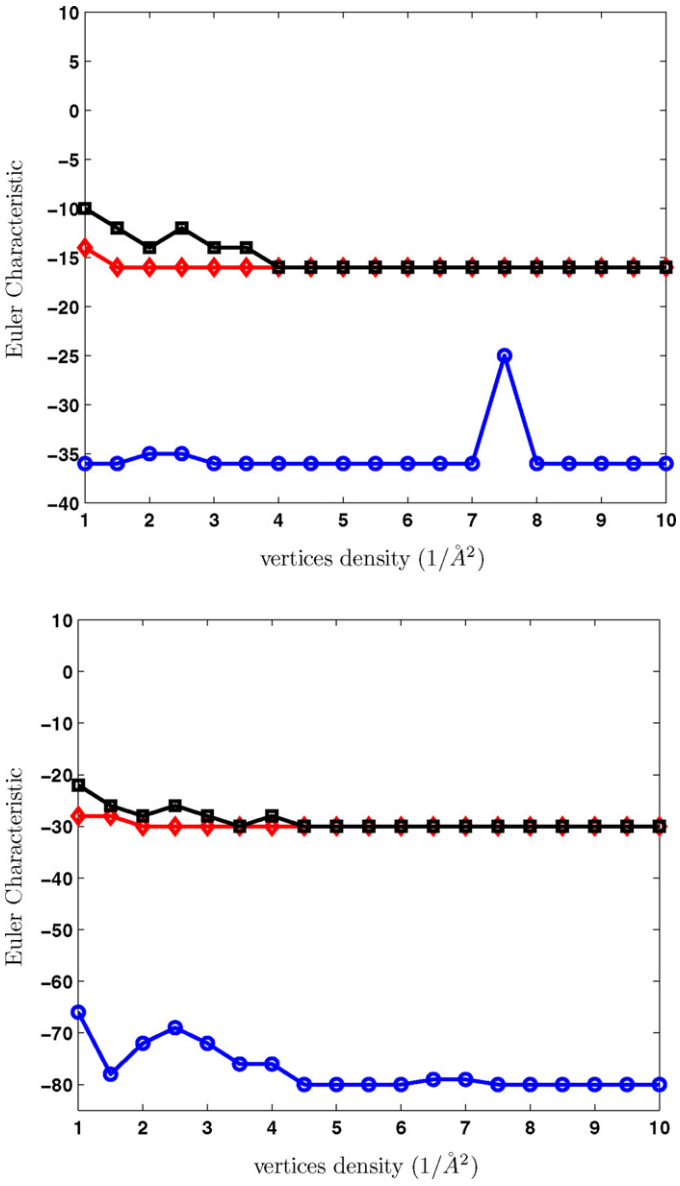

Fig. 5. Euler characteristics of surface meshes with different vertex densities for AChE monomer (left) and AChE tetramer (right) generated by MSMS (in blue color), the old version of TMSmesh (in black color) and the new version of TMSmesh (in red color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

This is reasonable because the mesh converges to the implicitly defined Gaussian surface with the increasing of the resolution and the new version of TMSmesh can find the structure smaller than the grid space through tracing the fold curves. Non-manifold errors in meshes produced by MSMS also cause the unstableness of the Euler characteristic when the vertex density increases. There are some discrepancies between the Euler characteristics calculated with TMmesh and MSMS meshes, which is due to the different definitions of molecular surface used in the mesh tools.

For the considerable large molecule such as 3K1Q (see Table 1), the surface meshes generated by the improved version of TMSmesh can be used to generate surface conforming volume mesh by volume meshing software TetGen [31] directly. However the meshes produced by MSMS and the old version of TMSmesh can not. Finally, we show a generated mesh of an ion channel protein, connexin. Connexin is an open hexametric channel that has a wide range of biological implications [32]. The pqr structure used in this

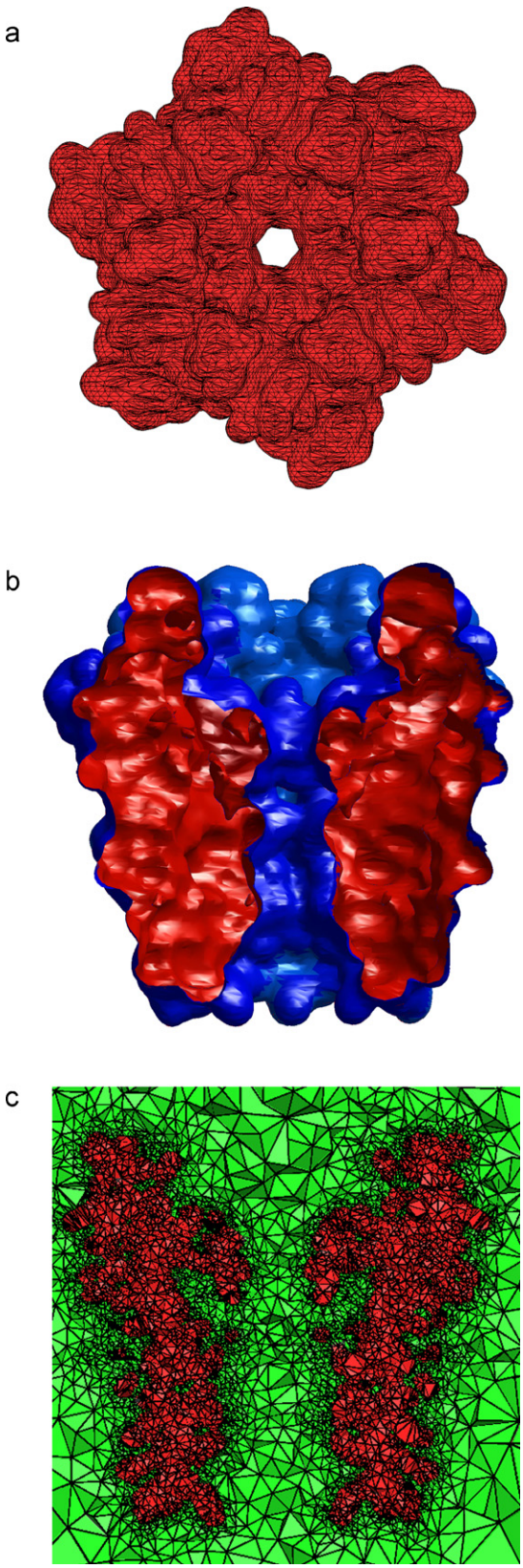

Fig. 6. (a) Shows surface mesh for ion channel connexin generated by TMSmesh. (b) Shows a cross section of surface mesh of the ion channel. The inside face of the surface is colored in red. The outside face is colored in blue. (c) Shows a cross section of the corresponding surface conforming volume mesh generated by TetGen [31]. The body of protein is in red.

calculation has total 1296 residues and 19,884 atoms. A surface triangular mesh and a volume mesh conforming the surface mesh are shown in Fig. 6. The channel pore is clearly represented in the surface mesh, and its detailed topology is correctly preserved, which is crucially important for channel permeation studies.

4. Conclusion

We have described an improved version of TMSmesh for meshing Gaussian molecular surface. The new meshing method divides the Gaussian surface into single valued pieces along each of x, y, z directions by tracing the extreme points along the fold curves on the surface. The new version of TMSmesh is shown as a robust tool for meshing Gaussian surface for large biomolecules and the produced meshes are manifold without intersections. Moreover, the mesh can be directly used (without modifications/corrections etc.) to generate corresponding surface conforming volume meshes.

Acknowledgments

M.X. Chen was supported by the Collegiate NSF of Jiangsu Province (No. 11KJB110010) and China NSF (No. NSFC11001062), B. Tu and B.Z. Lu were supported by the State Key Laboratory of Scientific/Engineering Computing, the National Center for Mathematics and Interdisciplinary Sciences, Chinese Academy of Sciences, 863 Program (2012AA020403), and the China NSF (Nos. NSFC10971218, NSFC11001257).

References

[1] M.X. Chen, B.Z. Lu, TMSmesh: a robust method for molecular surface mesh generation using a trace technique, Journal of Chemical Theory and Computation 7 (2011) 203–212.
[2] B.Z. Lu, Y.C. Zhou, M.J. Holst, J.A. McCammon, Recent progress in numerical methods for the Poisson–Boltzmann equation in biophysical applications, Communication in Computational Physics 3 (2008) 973–1009.
[3] B. Lee, F. Richards, The interpretation of protein structures: estimation of static accessibility, Journal of Molecular Biology 55 (1971) 379–384.
[4] F.M. Richards, Areas, volumes, packing and protein structure, Annual Review in Biophysics and Bioengineering 6 (1977) 151–176.
[5] H. Edelsbrunner, Deformable smooth surface design, Discrete & Computational Geometry 21 (1999) 87–115.
[6] P.W. Bates, G.W. Wei, S. Zhao, Minimal molecular surfaces and their applications, Journal of Computational Chemistry 29 (2008) 380–391.
[7] M.L. Connolly, Analytical molecular surface calculation, Journal of Applied Crystallography 16 (1983) 548–558.
[8] M.L. Connolly, Solvent-accessible surfaces of proteins and nucleic acids, Science 221 (1983) 709–713.
[9] A. Nicholls, R. Bharadwaj, B. Honig, Grasp: graphical representation and analysis of surface properties, Biophysical Journal 64 (1995) 166–167.
[10] R.J. Zauhar, Smart: a solvent-accessible triangulated surface generator for molecular graphics and boundary element applications, Journal of Computer-Aided Molecular Design 9 (1995) 149–159, 10.1007/BF00124405.
[11] H. Edelsbrunner, E.P. Mucke, Three-dimensional alpha shapes, ACM Transactions on Graphics 13 (1994) 43–72.
[12] M. Sanner, A. Olson, J. Spehner, Reduced surface: an efficient way to compute molecular surfaces, Biopolymers 38 (1996) 305–320.
[13] Y.N. Vorobjev, J. Hermans, SIMS: computation of a smooth invariant molecular surface, Biophysical Journal (1997) 722–732.
[14] J. Ryu, R. Park, D.S. Kim, Molecular surfaces on proteins via beta shapes, Computer-Aided Design 39 (2007) 1042–1057.
[15] T. Can, C.I. Chen, Y.-F. Wang, Efficient molecular surface generation using level-set methods, Journal of Molecular Graphics and Modelling 25 (2006) 442–454.
[16] M. Chavent, B. Levy, B. Maigret, Metamol: high-quality visualization of molecular skin surface, Journal of Molecular Graphics and Modelling 27 (2008) 209–216.
[17] H. Cheng, X. Shi, Quality mesh generation for molecular skin surfaces using restricted union of balls, Computational Geometry 42 (2009) 196–206.
[18] W. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, Computers & Graphics 21 (1987) 163–169.
[19] T. Ju, F. Losasso, S. Schaefer, J. Warren, Dual contouring of hermite data, ACM Transactions on Graphics 21 (2002) 339–346.
[20] Y. Zhang, G.X. Ch, R. Bajaj, Quality meshing of implicit solvation models of biomolecular structures, Computer Aided Geometric Design 23 (2006) 510–530.
[21] A. Hilton, A.J. Stoddart, J. Illingworth, T. Windeatt, Marching triangles: range image fusion for complex object modelling, IEEE International Conference on Image Processing (1996) 381–384.
[22] E. Hartmann, A marching method for the triangulation of surfaces, The Visual Computer (1998) 95–108.
[23] T. Karkanis, J. Stewart, Curvature dependent triangulation of implicit surfaces, IEEE Computer Graphics and Applications (2001) 60–69.
[24] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry: Algorithms and Applications, second edition, Springer-Verlag, Berlin, 2000.
[25] B.S. Duncan, A.J. Olson, Shape analysis of molecular surfaces, Biopolymers 33 (1993) 231–238.
[26] J.F. Blinn, A generalization of algebraic surface drawing, ACM Transactions on Graphics 1 (1982) 235–256.
[27] M.R. Mcgann, H.R. Almond, A. Nicholls, A.J. Grant, F.K. Brown, Gaussian docking functions, Biopolymers 68 (2003) 76–90.
[28] J.A. Grant, M.A. Gallardo, B.T. Pickup, A fast method of molecular shape comparison: A simple application of a Gaussian description of molecular shape, Journal of Computational Chemistry 17 (1996) 1653–1666.
[29] J. Weiser, P. Shenkin, W. Still, Optimization of Gaussian surface calculations and extension to solvent-accessible surface areas, Journal of Computational Chemistry 20 (1999) 688–703.
[30] Z. Yun, P. Matthew, A. Richard, What role do surfaces play in gb models? a new-generation of surface-generalized born model based on a novel gaussian surface for biomolecules, Journal of Computational Chemistry 27 (2005) 72–89.
[31] H. Si, K. Gaertner, Meshing piecewise linear complexes by constrained delaunay tetrahedralizations, in: Proceedings of the 14th International Meshing Roundtable, Springer, 2005, pp. 147–163.
[32] S. Maeda, S. Nakagawa, M. Suga, E. Yamashita, A. Oshima, Y. Fujiyoshi, T. Tsukihara, Structure of the connexin 26 gap junction channel at 3.5 angstrom resolution, Nature 458 (2009) 597–602.
[33] D. Zhang, J.A. McCammon, The association of tetrameric acetylcholinesterase with colq tail: a block normal mode analysis, PLoS Computational Biology (2005) 484–491.
[34] H. Jin, A.C. Kelley, D. Loakes, V. Ramakrishnan, Structure of the 70s ribosome bound to release factor 2 and a substrate analog provides insights into catalysis of peptide release, Proceedings of the National Academy of Sciences of the United States of America (2010) 8593–8598.
[35] R.J. Kuhn, W. Zhang, M.G. Rossmann, J.C. Sergei, V. Pletnev, E. Lenches, C.T. Jones, S. Mukhopadhyay, E.G.S. Paul, R. Chipman, T.S. Baker, J.H. Strauss, Structure of dengue virus: implications for flavivirus organization, maturation, and fusion, Cell (2002) 715–725.