Molecular Surface Maps

Michael Krone*, Florian Frieß*, Katrin Scharnowski, Guido Reina, Silvia Fademrecht,
Tobias Kulschewski, Jürgen Pleiss, and Thomas Ertl

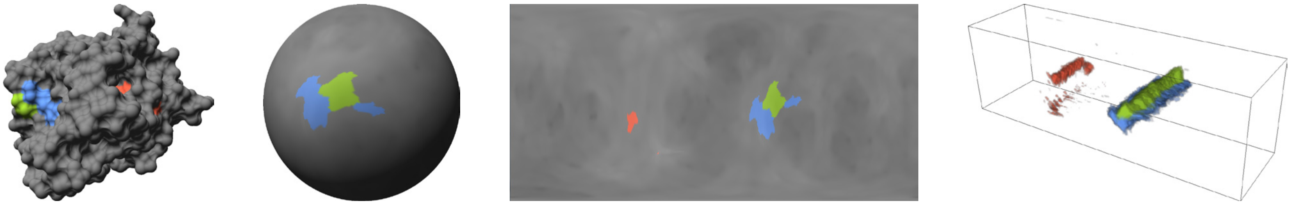

Fig. 1. Results of our Molecular Surface Maps visualization for a simulated protein (PDB ID: 1TCA). From left to right: The original Molecular Surface (Solvent Excluded Surface) is first mapped to the Molecular Surface Globe, from which the Molecular Surface Map is obtained using the cylindrical equal-area Hobo-Dyer map projection. The colors (green, blue, red) show binding sites for ligands on the surface. The gray values correspond to the deformation of the mesh (i.e., the path length traveled by each vertex of the original molecular surface mesh; darker is longer). The rightmost image shows a Space-time Cube of Molecular Surface Maps for all time steps of the simulation, illustrating the temporal development of the binding sites (i.e., their availability at the surface over time).

Abstract—We present Molecular Surface Maps, a novel, view-independent, and concise representation for molecular surfaces. It transfers the well-known world map metaphor to molecular visualization. Our application maps the complex molecular surface to a simple 2D representation through a spherical intermediate, the Molecular Surface Globe. The Molecular Surface Map concisely shows arbitrary attributes of the original molecular surface, such as biochemical properties or geometrical features. This results in an intuitive overview, which allows researchers to assess all molecular surface attributes at a glance. Our representation can be used as a visual summarization of a molecule's interface with its environment. In particular, Molecular Surface Maps simplify the analysis and comparison of different data sets or points in time. Furthermore, the map representation can be used in a Space-time Cube to analyze time-dependent data from molecular simulations without the need for animation. We show the feasibility of Molecular Surface Maps for different typical analysis tasks of biomolecular data.

Index Terms—Molecular Visualization, Maps, Cartography, Data Aggregation, Dimensionality Reduction, Space-time Cube

1 Introduction

Molecular surfaces are among the most widely used visual representations for the analysis of molecules, especially for the analysis of molecular simulations in biochemistry and structural biology. Different surface models show individual properties of the molecule [25]. Additional information can be mapped to the surface using color, for example to show physico-chemical properties of the underlying atoms.

We present Molecular Surface Maps, a novel, view-independent representation for molecular surfaces and their attributes. The aim was to create a visualization that shows the physico-chemical properties as well as the topography of the molecular surface in a single image that summarizes the surface and resolves common problems inherent to three-dimensional visualizations such as view-dependency and occlusion. To achieve this goal, we transferred the world map metaphor to molecular surface visualization. That is, like in cartography, the whole geometry of the molecular surface is mapped to a two-dimensional image. Thus, the analyst can get a quick overview of the whole surface and identify interesting surface features for further investigation.

Following the geography metaphor, the molecular surface is first morphed to a spherical intermediate, which we call the Molecular Surface Globe. From this globe, the Molecular Surface Map is derived using established map projection methods from cartography (see Figure 1). This creates an intuitive overview of the molecular surface, since we can safely assume that all users are familiar with world maps and can relate to the depiction. To ensure this for all stages of the mapping, it is important to use a comprehensible morphing of the original surface to the spherical intermediate. However, molecular surfaces can have an arbitrary genus, due to channels that run through the molecule or handle-like structures. This poses a problem for the morphing of the molecular surface to the spherical intermediate, which is genus zero. In contrast to most prior methods discussed in Section 2 that only work for molecular surfaces of genus zero, we address this issue by first searching for all holes and closing all of them so that the outer interface is maintained. Note that Molecular Surface Maps are not intended to replace classical 3D visualization of molecular surfaces but rather to complement it. The map can be used as a first step in the analysis workflow to get an overview and identify areas of interest, which can be analyzed in detail during the next step using classical molecular surfaces. Beside the use as an overview visualization, Molecular Surface Maps can be applied to compare the interfaces—that is, the surface properties—of different molecules, which is another common analysis task in biochemistry. The summarization of the interface provided by the map can alleviate the identification of interesting molecules from a larger ensemble of data sets based on surface properties.

Our Molecular Surface Maps algorithm was developed in close collaboration with domain experts from biochemistry. We implemented an application prototype that was provided to the domain experts for evaluation to make sure that it supports them in their daily work.

* MK & FF contributed equally and should be regarded as joint first authors.

- M. Krone, F. Frieß, K. Scharnowski, G. Reina, and T. Ertl are with the Visualization Research Center (VISUS), University of Stuttgart, Germany. E-mail: {michael.krone|florian.friess|katrin.scharnowski|reina|ertl}@visus.uni-stuttgart.de.
- S. Fademrecht, T. Kulschewski, and J. Pleiss are with the Institute of Technical Biochemistry (ITB), University of Stuttgart, Germany. E-mail: {silvia.fademrecht|tobias.kulschewski|juergen.pleiss}@itb.uni-stuttgart.de.

Manuscript received 31 Mar. 2016; accepted 1 Aug. 2016. Date of publication 15 Aug. 2016; date of current version 23 Oct. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2016.2598824

Contributions: We developed a novel visualization for molecular data that maps the complex molecular surface to a two-dimensional representation. Our method applies the well-known world map metaphor to molecular surface visualization. One major contribution of our Molecular Surface Map method compared to previous methods is that it can handle molecular surfaces of arbitrary genus. Molecular Surface Maps can also be applied to time-varying data originating from molecular simulations. Since they are view-independent, they can be used to construct a Space-time Cube that allows the user to investigate the temporal development of surface features for the whole simulation without the need for animation. Finally, we show the feasibility of our Molecular Surface Maps with use cases for the analysis of different, real-world data sets from a user's perspective.

2 Related Work

Our visualization method builds upon molecular surfaces (a comprehensive overview of molecular surface definitions can be found in the recent report by Kozlikova et al. [25]). In particular, we use the Solvent Excluded Surface (SES) [43], which is one of the most useful and, therefore, most commonly used molecular surface representations. The SES is the interface of a molecule with respect to a spherical probe. The radius of this probe approximates a solvent molecule of a certain size. That is, small gaps between atoms that are not reachable by the probe are closed (see Figure 1, left). The SES can be computed analytically [10]. For rendering, the SES can either be triangulated (see, e.g., [47]) or directly rendered via ray casting of the implicitly defined surfaces patches (cf. [27, 34]).

Molecular Surface Maps transfer the commonly used geographical map representation to molecular visualization by creating a 2D map of the molecular surface that shows the molecular interface. Thus, they are comparable to other 2D representations for molecular structures. A widely used non-spatial depiction for proteins is the sequence diagram. It shows a simple string of the amino acids that form the protein using a one-letter code. Usually, sequence diagrams are enriched with additional information like secondary structure [44] or information about binding sites. Another 2D visualization that is commonly used for proteins is the Ramachandran plot [42], which shows the backbone dihedral angles ψ and φ. It is used to analyze and compare conformations and to estimate secondary structures. The Protein Data Bank [6] provides both sequence diagrams and Ramachandran plots for all recorded protein structures. While these two visualizations are useful to analyze the structure of a protein, they are not correlated to its interface. That is, they show fundamentally different aspects of the visualized molecule compared to our Molecular Surface Maps.

In general, creating 2D representations from higher-dimensional data has been a topic of active research for quite some time. Usually, the low-dimensional representation is just a similarity space where data points are arranged to obey a distance matrix containing some high-dimensional dissimilarity measure as best possible. Multidimensional scaling is one such technique, which iteratively minimizes a loss function [8]. This is extremely costly and has been improved several times over the years. Other techniques include FastMap, which uses heuristics to find significant high-dimensional axes to project onto [16], or Locally Linear Embedding, which explicitly tries to preserve neighborhood [45]. Specific 2D representations for molecular surfaces via Self-Organizing Maps (SOMs) have also been proposed. Neurons arranged on a torus surface (plane with periodic boundary conditions) have been used to produce a 2D mapping [2]. A spherical SOM has been used as an alternative topology for lower distortion, while still being prone to occlusion [19].

Our Molecular Surface Map algorithm projects a molecule surface onto a 2D map. That is, it can essentially be seen as a parametrization of the molecular surface. Surface parametrization is crucial in several applications, such as mesh morphing, texture mapping, or remeshing of surfaces. A lot of work has been published concerning this topic, therefore, we refer the reader to the extensive survey of Hormann et al. [23]. For parametrization, a mapping to a simpler intermediate domain—such as a plane or a sphere—is often used [46]. An overview of planar parametrization, as for example used in texture mapping, was given by Wei et al. [59]. In general, for planar parametrization, the mesh has to be cut to the equivalent of a disc or decomposed into a number of separate discs [5, 17], which can then be parametrized individually. Energy minimization can be used to reduce texture distortion on the surface [35]. Mapping of genus zero surfaces to a sphere requires no cutting [40]. Other methods use a direct mapping, which does not require an intermediate domain [26, 51]. These methods, however, often focus on a smooth morphing between meshes and not on a complete mapping of one mesh onto the other one. Shape correspondence (i.e., the problem to establish a mapping between two given shapes) is closely related to surface parametrization [54]. Early work used deformable models for segmentation [55].

In the field of computational chemistry, molecules are often compared based on local surface attributes [22, 11]. Baum et al. [4] aligned molecular surfaces using an iterative closest point scheme to establish correspondence between the SES vertices. Scharnowski et al. [48] used a deformable model approach to establish shape correspondence between two molecular surfaces for comparison. They applied a diffusion-based external force—the Gradient Vector Flow (GVF) introduced by Xu et al. [61]—for the morphing process to achieve a partial mapping of cavities or protrusions. Several works that establish a correspondence between a molecular surface and a sphere have been proposed. Postarnakevich et al. [39] also used deformable models to project surface properties to a sphere. A bijective mapping for triangulated molecular surfaces of genus zero onto a sphere that uses a parameterization based on spherical coordinates was developed by Rahi and Sharp [41]. Hass and Koehl [20] used a conformal mapping between triangulated molecular surfaces of genus zero and a sphere to measure the globularity of the molecule and to measure shape similarities of molecular surfaces. Polyansky et al. [38] presented a method than creates 2D maps of helical dimers based on the distance and the rotation angle with respect to the helical axis. That is, their method cannot be applied to arbitrary molecules. Structuprints by Kontopoulos et al. [24] projects surface points onto a sphere along a straight line from the center of the molecule. Surface points can, therefore, overlap on the sphere—especially for surfaces of higher genus—while other parts can be devoid of surface points. While the resulting map creates a 'fingerprint' of the molecule as intended by the authors, it does not give a complete overview of the interface.

As mentioned in the introduction, Molecular Surface Maps can be used in Space-time Cubes. Space-time cubes are a commonly used method to visualize temporal data—that is, the development of values in a 2D domain—and were originally designed for geographic data [18]. For a recent review of Space-time Cubes and their application, please refer to Bach et al. [3].

3 Molecular Surface Map Algorithm

The algorithm to create the Molecular Surface Map of a molecule consists of the following four steps, which will be detailed below:

1. Compute the underlying molecular surface.
2. Convert molecular surfaces of genus *n* to genus zero.
3. Compute the Molecular Surface Globe: Map the molecular surface to a sphere.
4. Create the Molecular Surface Map from the Molecular Surface Globe using map projection.

Compute the underlying molecular surface

The input for our Molecular Surface Map algorithm is a triangulated molecular surface mesh. As mentioned in Section 2, we focus on the Solvent Excluded Surface, since it is a viable choice for many different analysis tasks. In theory, however, our method is independent of the input surface. That is, it could be used with all other kinds of molecular surfaces that can be represented by a mesh like Metaballs [7] or Skin Surface [14].

Convert molecular surfaces of genus *n* to genus zero

In Section 2, we discussed algorithms that map a molecular surface of genus zero to a sphere (e.g., [41, 20]). Especially proteins, however, often have a surface of genus *n* due to channels that cross the protein. Mapping a surface of genus *n* to a sphere is not straightforward. Therefore,

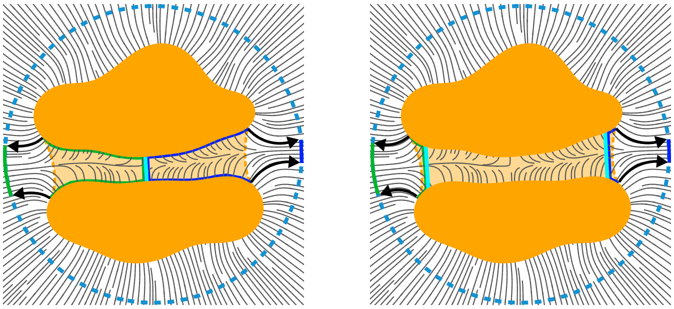

Fig. 2. Channels through a protein can be treated in two ways: The molecular surface mesh can be bisected at the center of the channel (left). Afterwards, the cyan cut is closed from both sides (blue and green surfaces). Alternatively, two cuts can be placed at the entrances of the channel (right). Here, the channel itself will be removed.

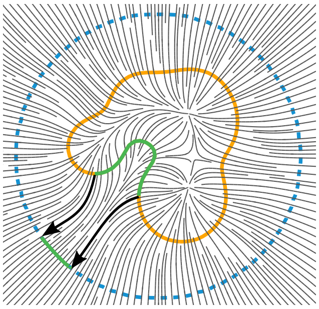

Fig. 3. The gradient vector flow (GVF) field mapping the molecular surface (orange) to the target sphere (blue, dashed). The distance between the sphere and the molecule is exaggerated for clarity. When using the force-directed approach without internal forces, the green cavity is mapped to a comparably small area on the globe.

the second step in the algorithmic pipeline is to identify and close all channels prior to the mapping in order to obtain a surface of genus zero. A channel can be closed by placing a cut at an arbitrary position in the channel and closing the resulting holes. The placement of the cut that removes a channel influences the mapping onto the sphere. Ideally, the cut would be placed halfway through the channel. The resulting two halves of the channel can then be mapped to the sphere (see Figure 2, left). However, since they are very deep, they will either be mapped to a very small area on the sphere or the surrounding regions will be distorted. In both cases it will be harder to discern the information on the final map. This issue is also present in geographic cartography: Mountains or valleys are flattened onto the globe by simply projecting them along the radial axis. That is, distances between sample points are also not preserved on the globe and their area will consequently be distorted too. However, the height of a mountain—or the depth of a valley, respectively—is very small compared to the radius of the earth, therefore, it is less obvious than for a molecular surface. An alternative that can remedy this issue for molecular surfaces is to use two cuts that are close to the entrances of the channel. In this case, the entrances will be closed and the channel itself will be removed. The resulting valleys are comparably small and can be mapped with lower distortion (see Figure 2, right). The drawback of this method is that information within the channel will be lost on the globe and the final map. A combination of both methods can be used as a tradeoff: The inner parts of long channels are removed to lower the distortion, while short tunnels are bisected and mapped completely.

Compute the Molecular Surface Globe The next step in the algorithmic pipeline is to map the molecular surface to a sphere, resulting in the Molecular Surface Globe. The sphere that forms the Molecular Surface Globe is the bounding sphere of the molecular surface. Since the molecular surface has genus zero after removing all channels, the methods discussed in Section 2 can be applied. An important requirement of the mapping is that it should be intuitively understandable by a human observer. Distortions should be kept low, that is, the location of an arbitrary point on the molecular surface with respect to other surface points should be similar on the globe. Consequently, the areas on the molecular surface and the globe will correspond to each other as well. Thus, force-directed approaches like the one of Postarnakevich et al. [39] seem appropriate. They morph a tessellated sphere to a target molecular surface. We used a similar approach but morph the molecular surface to the sphere. This direction was also used in the method of Hass and Koehl [20]. For the computation of the forces that pull the molecular surface towards the sphere, the sphere can be represented implicitly by its center and radius. This allows for higher accuracy, since an explicit target shape would require voxelization and, hence, introduce additional discretization errors. We use a deformable model [55] to map a point on the molecular surface to the sphere. Deformable models distinguish between external and internal forces that act on the vertices of the mesh, which gives us additional control over the way the input surface deforms. To obtain the external forces, we use the modified GVF that was also used by Scharnowski et al. [48]. This algorithm creates an external force field by diffusing both the individual components of the normals of the source shape (i.e., the molecular surface) and of the target one (i.e., the sphere). An illustration of the GVF field can be found in Figure 3.

A drawback of the aforementioned force-directed approach is that the quality of the mapping highly depends on the input parameters. An alternative to the force-directed approach is the method presented by Rahi and Sharp [41]. Their algorithm re-parametrizes the vertices of the original molecular surface by assigning spherical coordinates θ and ϕ to every vertex in order to map them to a sphere. The first step is to identify the north pole and south pole. Rahi and Sharp proposed to use the two vertices with the biggest Euclidean distance. In our case, the original orientation of the molecule can be determined by the domain scientists, that is, it is meaningful and must not be changed. Therefore, we chose the vertices that are closest to the north-south axis of the bounding sphere as poles. In the second step, neighbors of the north pole are assigned the value $z = +Z$ and neighbors of the south pole are assigned the value $z = -Z$, with Z being any arbitrary value greater than zero. To all other vertices, the average values of their neighbors assigned iteratively. Using the inverse Mercator projection—with the z values as input—the θ values can be determined. The ϕ values are obtained by creating a boundary meridian between the two poles and iteratively applying the average values of neighboring vertices to all vertices. This meridian is found starting from the north pole and following the triangle edges with the steepest decent until the south pole is reached.

Regardless of the chosen mapping algorithm, morphing the molecular surface to the sphere inevitably generates a non-uniform point distribution on the sphere. This is due to the complex shape of the molecular surface, which has protrusions (“mountains”) and cavities (“valleys”). This relates to the issue mentioned above that also exists for geographical globes of the earth: If a cavity is flattened onto the sphere, either the distances between the original sample points will be reduced or the points surrounding the cavity will be forced away. The same is true for a protrusion. Our aim is to maintain the original distances between the sampled points (e.g., the vertices of the molecular surface mesh) as good as possible to create a meaningful and intuitive transformation of the surface mesh vertices onto the sphere. For both the force-directed approach as well as the parameter-based method, this results in a pareto-optimal mapping where points within dense regions try to relax without pushing the surrounding points too far away. Consequently, the areas of a feature on the surface will not exactly correspond to its original area on the molecular surface. A comparison of the force-directed approach and the parameter-based one is given in Section 6. Furthermore, the original topography of the molecular surface can also be conveyed through color (see Section 4.1).

Create the Molecular Surface Map In the final step of our algorithmic pipeline, an arbitrary map projection can be used to transform

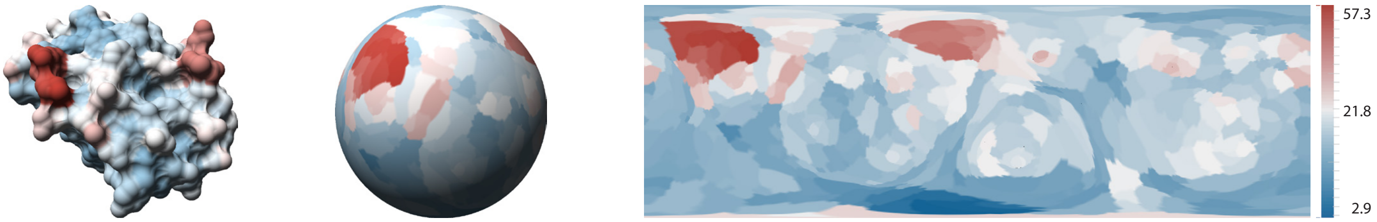

Fig. 4. Molecular Surface Globe and Map of a small protein colored by temperature factor (PDB ID: 1RWE). The legend to the right shows the color gradient mapped to the temperature factor values. The map was created using the force-directed method and Lambert equal-area projection.

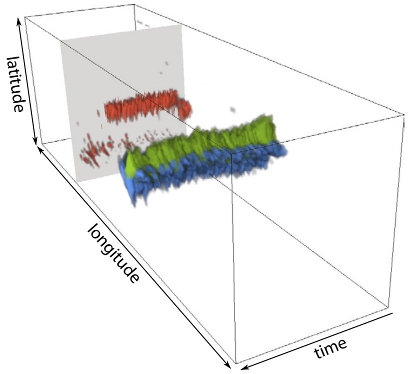

Fig. 5. Space-time Cube showing the evolution of binding sites for a protein simulation. Towards the end of the simulation, the red binding site vanishes. That is, the protein has deformed so that it is not accessible any more. The gray clipping plane was added as a depth cue.

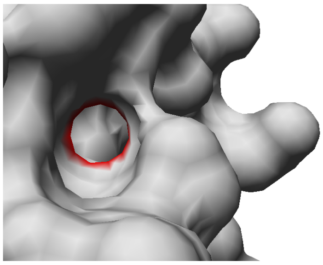

Fig. 6. Tunnel loop (red) in the SES of a protein found by the method of Dey at al. [13]. The loop denotes a hole in the surface that has to be closed to obtain a surface of genus zero.

the sphere to a map. A comprehensive overview of map projections along with the respective equations can be found in the report by Snyder and Voxland [53]. Although the original areas of the molecular surface are not necessarily preserved on the Molecular Surface Globe, a cylindrical equal-area map projection like Lambert would be a reasonable choice (see Figure 4). The rationale for this is that we do not want to introduce an even larger error through the map projection. Several alternative equal-area projections that are horizontally compressed versions of the Lambert projection have been proposed and can be used if a different aspect ratio is desired, for example Hobo-Dyer (see Figure 1) or Gall-Peters (see Figure 13). Another option would be to use a cylindrical equirectangular map projection like Plate Carrée, which is equidistant along the meridians. Both types of projections create rectangular maps, which are not only commonly used in atlases of the world but also visually similar to the variant of the Mercator projection used by Google Maps (`http://maps.google.com`), which makes it easier for users to relate to the resulting Molecular Surface Map.

3.1 Extension to Time-dependent Data

Molecular Surface Maps are a 2D representation of the molecular surface, which allows us to represent dynamic data sets using Space-time Cubes. Space-time Cubes are commonly used in geovisualization to show the temporal evolution of data points or features on a map [18]. For each time step of the simulation, the Molecular Surface Map is computed and stored as slice of a volume. Here, the molecule has to be previously aligned for all time steps, e.g., by applying the commonly used RMSD alignment. The recent review by Bach et al. [3] gives a taxonomy of commonly used operations. The usefulness of Space-time Cubes for Molecular Surface Maps depends largely on the map coloring, which in turn is determined by the analysis task.

A common task in structural biology is to monitor the presence of pockets or the availability of binding sites throughout a molecular simulation. For this, the volume of the Space-time Cube can be visualized using the *oblique flattening* operation, that is, as a perspectively projected cube. Uninteresting parts can be removed using the *filtering* operation. In the case of binding sites or pockets, the neutral color has to be filtered, that is, only the surface features will remain (see Figure 1). Features can either be visualized using volume rendering [30] or by extracting isosurfaces. An isosurface extraction would implement the *aggregation* operation. The analyst can now see how a surface feature evolves and whether it is persistent in time. An example for this can be seen in Figure 5, where the red binding site vanishes at some point in time. To prevent occlusion during analysis, camera adjustment has to be supported (*rotation* and *shifting* operations).

4 Description of the Application

We implemented an application that allows users to create Molecular Surface Maps of arbitrary molecules. The application can load molecules stored in the popular PDB file format [6]. It also supports time-dependent data stored as trajectories in the GROMACS XTC file format[1]. Our application allows users to adjust all parameters of the algorithmic pipeline that computes the Molecular Surface Map.

First, the Solvent Excluded Surface (SES) has to be computed. As explained in Section 2, the SES is defined by a spherical probe that rolls over the union of spheres that represent the atoms of a molecule [43]. Consequently, the SES shows the molecular interface with respect to another probe-sized molecule. The users can set the radius of the probe sphere for the SES computation as well as the tessellation level for the triangulation of the molecular surface.

The next step is to remove long channels from the molecule in order to achieve a better mapping. This step is optional. Our application uses an extended version of the method presented by Krone et al. [29, 28], which uses Ambient Occlusion (AO) to identify channels in proteins. This approach was introduced by Borland [9]. A similar method that also samples the neighborhood for occluding geometry to find channels was used by Weisel et al. [60]. The user can influence the channel extraction by adjusting the threshold value for the AO. For a high threshold, only the innermost part of a longer channel will be detected, whereas a low threshold will lead to the detection of the whole channel. If the AO threshold is set to zero, the channel removal step is skipped. Subsequently, all identified channels are removed and the newly arising holes in the surface are closed.

[1] GROMACS Online Reference – File formats: `http://manual.gromacs.org/online/files.html` (last accessed 03/25/16).

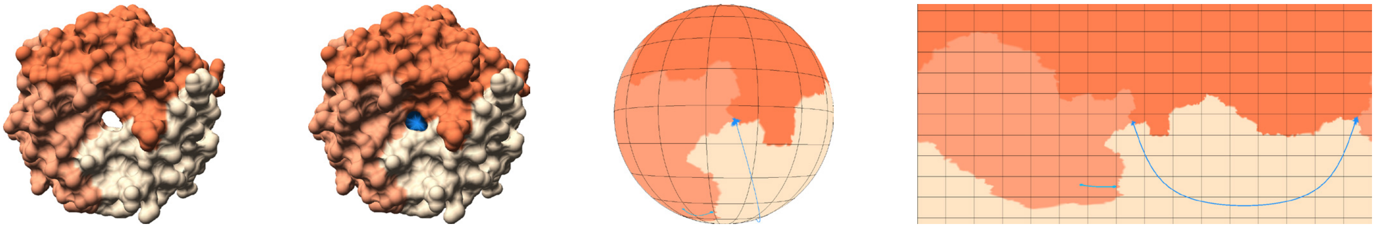

Fig. 7. Molecular Surface Map of a protein with a channel (PDB ID: 2BT9, map created using the parameter-based method and Plate Carrée projection). The coloring shows the three amino acid chains that form the protein. The entrances of the two channels are visible as blue areas on the map. The channel connectivity is illustrated using geodesic lines in the same color as the entries that are connected by the original channel.

After the channel removal, the surface can still be of genus $n > 0$. Therefore, in the next step, all remaining channels or handle-like structures are found. Our algorithm uses the method of Dey et al. [13], which finds minimal tunnel loops in a surface mesh using a Reeb graph. Figure 6 shows an example of a tunnel loop in the SES. The mesh is cut open at these tunnel loops and closed from both sides to obtain a surface of genus zero. In contrast to the optional channel removal, no surface information is lost since only a one triangle wide ring of the SES is removed for each tunnel loop. No user input is required for this step.

The user can choose between the force-directed method and the parameter-based method described in Section 3 to morph the SES mesh onto the Molecular Surface Globe (for more details about the difference of the methods see Section 6). The force-directed method requires the user to set the ratio between external and internal forces for the morphing as well as the minimum displacement for the vertices (which serves as a convergence criterion, see Section 5). The parameter-based method requires no input. Both methods maintain the topology of the mesh, that is, local neighborhoods will be preserved.

Currently, the application offers two types of cylindrical map projections: equal-area (Lambert, Hobo-Dyer) and equirectangular (Plate Carrée). The user can interactively switch between these projections.

The equator as well as the prime meridian are defined by the original orientation of the molecule as stored in the input file. The user can choose to display a grid of latitude and longitude lines on the globe and the map. To facilitate the orientation, equator and prime meridian can be shown in different colors than the rest of the lines. Initially, the map is centered on $Lat = 0°, Lon = 0°$. To enable a comprehensive analysis, the user can switch between the three main visualization modes: Solvent Excluded Surface, Molecular Surface Globe, and Molecular Surface Map. The user can also interactively change the viewport. The view on the Molecular Surface, the Globe, and the Map are always coherent. That is, the map will be shifted so that the center always matches the focal point of the 3D visualizations. This especially facilitates the analysis of surface regions that are near a pole and would be stretched on the map in the initial orientation (see Figure 8).

As mentioned above, the closing of holes through the molecular surface (channels and handle-like structures) modifies the surface. The newly created triangles that close the holes are colored individually to be clearly visible. All entries of a closed hole have the same color, so that users can easily see which entries are connected. To further facilitate the visual analysis, entries of a hole can additionally be connected using geodesic lines on the globe and on the map. These lines resemble the widely used airplane route metaphor and clearly illustrate which entries are connected. An example is shown in Figure 7.

Our application offers pre-defined color tables for all different color modes. However, users can also define their own color tables. Finally, our application offers the possibility to store the Molecular Surface Map as a PNG image. The Space-time Cube of a dynamic data set can be stored as an RGBA RAW volume file.

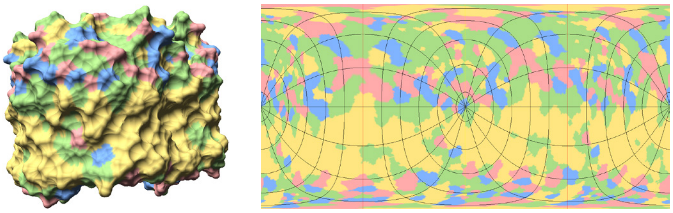

Fig. 8. Coloring by physico-chemical properties of the amino acids (PDB ID: 1AF6). The colors encode whether the amino acids are polar (green), hydrophobic (yellow), basic (blue), or acidic (red). The map was created looking along the pole axis. The lower part of the protein is the transmembrane domain, which contains mostly hydrophobic amino acids. This is also clearly visible in the lower half of the map, which is mostly yellow. (Parameter-based method, Hobo-Dyer projection)

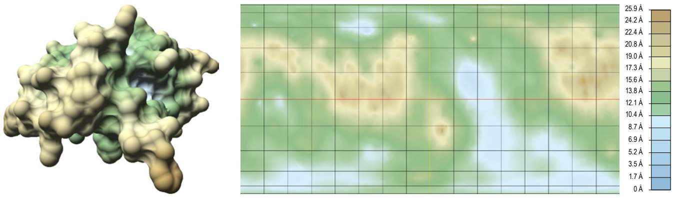

Fig. 9. Cartography-inspired coloring by elevation. The coloring illustrates valleys and mountains of the original protein surface (left, PDB ID: 1RWE) on the Molecular Surface Map (right, Hobo-Dyer projection).

4.1 Molecular Surface Map Coloring

As with geographical cartography, coloring is a vital part of our Molecular Surface Maps, since it shows the features of the projected molecule. Our application offers suitable map colorings for analysis tasks that use Molecular Surfaces Maps. Although our Molecular Surface Map application can be used with all kinds of molecules, the colorings focus on proteins, since we designed the map representation to be used in structural biology.

All common (bio-)chemical coloring schemes can be applied to Molecular Surface Maps, since the algorithm described above simply maps points on the molecular surface to the Molecular Surface Map. Coloring schemes that illustrate properties that influence the molecular interface are particularly useful. This includes hydrophobicity, temperature factor, or the electrostatic potential at the molecular surface. Since these properties are signified by a single value for each point on the molecular surface, a color gradient is typically used for rendering (see Figure 4). Categorizing colorings by physico-chemical properties of the amino acids are also useful (see Figure 8).

Another important property is the availability of binding sites, which consist of a set of amino acids that have to be exposed to the molecular surface in order to be accessible. For visualization, all atoms that belong to an amino acid of the same binding site are assigned the same color (see Figure 1). A related coloring mode shows the binding of solvent molecules to the molecular surface. For each point on the molecular surface, we search for the closest solvent of a specific type (e.g., water). If a solvent molecule is close to the surface (within 3 Å), it is classified as bound to the surface and the surface point is highlighted using color. For a simulation, the binding can be aggregated over time, so the surface coloring can show preferred binding sites of this solvent. An example is shown in Figure 13.

Color can also illustrate the topography (i.e., the shape) of the molecular surface. Our application includes a geography-inspired coloring that maps the elevation of a point to a color (see Figure 9). While geographic maps usually use the sea level as reference point, we simply determine the elevation as the distance to the center of mass of the molecule. An alternative is to use the length of the path that a point of the molecular surfaces has to travel until it reaches the Molecular Surface Globe (see Figure 1). This coloring can also be used to identify deformations when comparing different conformations of the same molecule. Another option is to highlight topographical features like cavities ("valleys") or protrusions ("mountains"). For example, available software like 3D-Surfer [33] can be used to extract all amino acids that form a cavity or protrusion, which can then be highlighted by color on the Molecular Surface Map. Another possibility to highlight the shape is to compute the Ambient Occlusion (AO) [64] of the original molecular surface and show the AO factors on the map.

5 Prototype Implementation

In this section, we give implementation details of our Molecular Surface Map application. The application is written in C++ using OpenGL for rendering and OpenCL to accelerate some of the computations.

For the computation of the SES, we use the well-established software MSMS by Sanner et al. [47]. It computes the SES analytically and generates a triangulated mesh of the surface. The inputs are the atoms of the molecule, their van der Waals radii, and the desired tessellation level for the surface mesh. MSMS is called directly by our application.

The first step is to check whether the input surface is already of genus zero using the Euler characteristic. If this is the case, the Molecular Surface Globe is created without any changes to the surface mesh. Otherwise, our application will have to close all holes of the SES mesh. Optionally, the user can choose to remove longer channels, which will be identified using the ambient occlusion (AO)-based channel detection method presented by Krone et al. [29]. As mentioned in Section 3, removing longer tunnels will reduce the distortion (see Figure 2). For each vertex of the mesh, an AO value is computed using a set of normal distributed rays. The number of rays that do not intersect the surface is counted and divided by the total number of rays. Based on the user-defined threshold for the AO value, we group adjacent faces together and count their unique borders with other groups. If a group has more than one unique border—that is, more than one entry—it is a channel and is removed from the surface mesh. The holes in the mesh that are created during this process are filled with a triangle fan. A tessellation shader is used to subdivide these new triangles so that they have roughly the same size as the existing ones. The computation of the AO values is parallelized using OpenCL. To increase the performance, we use an octree for the ray-triangle intersections.

After the optional removal of longer channels, the surface might still not be of genus zero since the AO-based detection does not reliably identify short channels or handle-like structures. Thus, the new surface is used as input for the method of Dey et al. [13], which identifies all holes. For each hole, it returns a sets of vertices that form a tunnel loop. For each of these loops, one ring of faces is removed from the mesh and the holes are again filled with tessellated triangle fans.

For the Molecular Surface Globe, the bounding sphere of the SES mesh is computed using the Bouncing Bubble algorithm [56]. This algorithm computes an approximate smallest enclosing sphere for a set of points. In our case, the points are the vertices of the SES mesh.

The next step in our computational pipeline depends on the method to morph the modified genus-0 SES onto the sphere that was selected by the user. For the force-directed approach, the first step is the computation of the external gradient vector flow (GVF) [61] force field. As mentioned in Section 3, we employ the deformation method by Scharnowski et al. [48], which uses not only the gradient of the target shape but also of the input shape. Therefore, the normals of the SES mesh and the ones of the sphere are sampled to a three-dimensional grid with grid spacing Δ_g. A default value of $\Delta_g = 1.0$ Å ensures subatomic detail. This grid is the input for the iterative diffusion process of the GVF, which generates the external force field for the following mesh morphing step. The internal forces, on the other hand, require only the computation of a mesh Laplacian. This can be achieved with the summation described by Shen et al. [52]. The computations for both the external forces and the internal forces were implemented in OpenCL and run in parallel on the GPU.

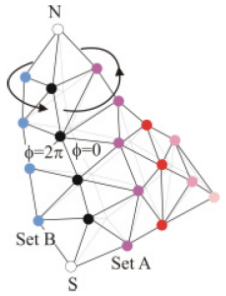

Fig. 10. Assignment of ϕ values for the parameter-based method [41]: For vertices in Set B (blue, $ID = 1$) the ϕ value of the boundary meridian vertices is 2π; for vertices in Set A (purple, $ID = 2$) the value is 0. The boundary meridian (black) connects north pole (N) and south pole (S).

We use the internal and external forces to morph the molecular surface mesh onto the sphere. That is, the vertex positions are iteratively moved according to the combined forces until convergence. For an appropriate step length, the forces are normalized and scaled by the grid spacing Δ_g. Additionally, we rescale the external forces by -0.5 every time the spherical target surface is crossed, which leads to quicker convergence and higher accuracy of the final vertex positions. The user can influence the morphing using the weighting parameter $\mu \in [0 \dots 1]$, which defines the ratio between external force and internal one. That is, external forces are scaled by μ and internal forces by $1.0 - \mu$. A higher value of μ leads to a less regular mesh, while a lower value of μ favors the regularizing internal forces. On the other hand, a lower value of μ increases the distortion of the mapping relation. For all data sets and images presented in this paper, we used a μ value of 0.5. The overall iteration process stops if the average displacement of all vertices is smaller than a minimum displacement d. We used a standard value of $d = 0.001$ Å for all data sets, which yielded good results. The final mesh is defined by the converged vertex positions while maintaining the triangle connectivity of the source mesh. The morphing process is also parallelized via OpenCL.

The original parameter-based algorithm by Rahi and Sharp [41] first identifies the two vertices with the biggest Euclidean distance and labels them north pole and south pole. Since our application maintains the original orientation of the molecule, we search for the vertices that are closest to the north-south axis that crosses the bounding sphere center and use them as north pole and south pole. Neighbors of the north pole are then assigned the value $z = +Z$ and neighbors of the south pole are assigned the value $z = -Z$, with $Z > 0$. Every other vertex is initialized with the value $z = 0$. Using the Gauss-Seidel relaxation they are updated using the unweighted average of their neighbors. After convergence, the θ values are computed using the inverse Mercator projection, with the z values as input. The boundary meridian can be detected by starting at a random neighbor of the north pole and following the path of steepest decent until a neighbor of the south pole is reached. The vertices of the meridian are assigned the $ID = 0$. Vertices on the left side are assigned the $ID = 1$ and vertices on the right side are assigned the $ID = 2$. Then the ϕ value of all non-meridian vertices is set to zero. The ϕ values are again propagated using the Gauss-Seidel relaxation using the unweighted average of their neighbors. If the meridian is approached by a vertex of with $ID = 2$, a ϕ value of 0 is propagated, if the meridian is approached by a vertex with $ID = 1$, a ϕ value of 2π is propagated (see Figure 10).

The map projections were implemented as GLSL shaders. The triangles on the Molecular Surface Globe are projected to the 2D domain of the map using either the equations for the cylindrical equal-area pro-

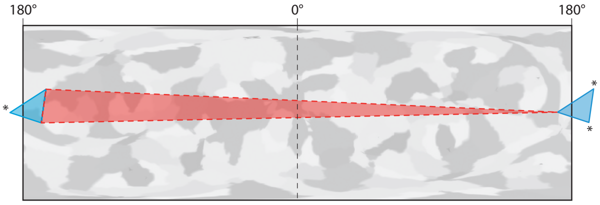

Fig. 11. Triangles that cross the antimeridian (180° meridian) on the globe will stretch across the entire Molecular Surface Map after projection (red). A correction step in the geometry shader duplicates the triangles and generate the correct vertex positions (*) for an artifact-free rendering (blue). Protruding parts of the triangles will be clipped.

jection (see equation 1) or the equirectangular one (see equation 2):

$$\text{Lambert:} \qquad x = \lambda - \lambda_0 \qquad y = \sin\phi \tag{1}$$

$$\text{Plate Carrée:} \qquad x = \lambda - \lambda_0 \qquad y = \phi \tag{2}$$

where λ is the longitude, λ_0 is the central meridian, and ϕ is the latitude. Equations 1 and 2 can be computed from the vertex positions in the vertex shader as follows:

$$\text{Lambert:} \qquad x = \frac{atan2(v_x, v_z)}{\pi} \qquad y = v_y \tag{3}$$

$$\text{Plate Carrée:} \qquad x = \frac{atan2(v_x, v_z)}{\pi} \qquad y = \frac{2 \cdot asin(v_y)}{\pi} \tag{4}$$

where (v_x, v_y, v_z) is the normalized vector from the bounding sphere center to the triangle vertex. Note that we scale the domain to $[-1, 1]$, since we render the result to an off-screen fragment buffer with the appropriate pixel ratio for the respective map projection. However, if we project whole triangles, wrong results occur for triangles that cross the antimeridian (180°). These triangles will stretch across the entire map, resulting in a defective rendering (z-fighting and jagged edges of the map). A correct map can be obtained by duplicating these triangles. For one triangle, vertices that cross the meridian are translated by 360°, for the other one, the vertices that do not cross the meridian are translated by -360°. The resulting triangles will automatically be clipped at the map boundary by the view frustum. Figure 11 shows the effect of this correction step, which was implemented as a geometry shader. Since our implementation uses GLSL shaders for the map projection that are only compiled at runtime, expert users can implement additional arbitrary map projections and use them in our application.

The final Molecular Surface Map can either be rendered directly or downloaded from the GPU and stored to an image file. For a Space-time Cube, the map for each time step is written to a binary RGBA volume file, which can for example be rendered using ParaView [21].

6 Results

We tested our application using protein data sets from the Protein Data Bank [6]. Table 1 shows the computation times (excluding time for SES computation via MSMS [47]). All tests were run on a desktop PC equipped with an Intel i7 3770 (3.4 GHz), 16 GB RAM, and an Nvidia Geforce GTX 760. Since the force-directed method was largely implemented using OpenCL, it has considerably lower runtimes than the parameter-based method, which is executed on the CPU.

As anticipated, the force-directed approach often creates a slightly more intuitive mapping of the SES for globular molecules. While the parameter-based method also produces very good results, it can lead to a less intuitive mapping of SES regions due to the relaxation of the ϕ and θ values. However, the difficulty of finding a good combination of parameters for the force-directed approach increases with the complexity of the underlying molecule. Especially for proteins with deep cavities, the method is very sensitive to changes in the ratio between external forces and internal forces. On the one hand, if the internal forces are weighted too high for proteins with deep cavities, the algorithm might not converge. On the other hand, if the internal forces are weighted too low, the triangles within the cavity will be mapped to a very small area, leading to high distortion. In contrast, the parameter-based method is able to produce a valid mapping even for very complex cases and requires no user-defined parameters.

Table 1. Comparison of the computation times of the force-directed method (FD) and the parameter-based one (PB) for different proteins. g_{SES} denotes the genus of the SES, g_{AO} the genus after removing long channels. Timings are denotes as follow: t_{AO}: long channel removal [29]; t_{TL}: tunnel loop computation [13]; t_{MSG}: Molecular Surface Globe computation; t_{total}: *total time.* All times are measured in seconds.

	#atoms	g_{SES}	t_{AO}	g_{AO}	t_{TL}	t_{MSG}	t_{total}
FD: 1RWE	816	0	–	0	–	1.29	1.70
PB: 1RWE	816	0	–	0	–	11.74	12.16
FD: 1TCA	2324	0	–	0	–	2.34	2.82
PB: 1TCA	2324	0	–	0	–	25.25	25.43
PB: 3ZHB	4016	4	20.77	3	5.29	71.99	98.07
FD: 1AF6	10,050	8	81.91	7	21.22	14.65	116.41
PB: 1AF6	10,050	8	81.91	7	21.22	237.35	341.26

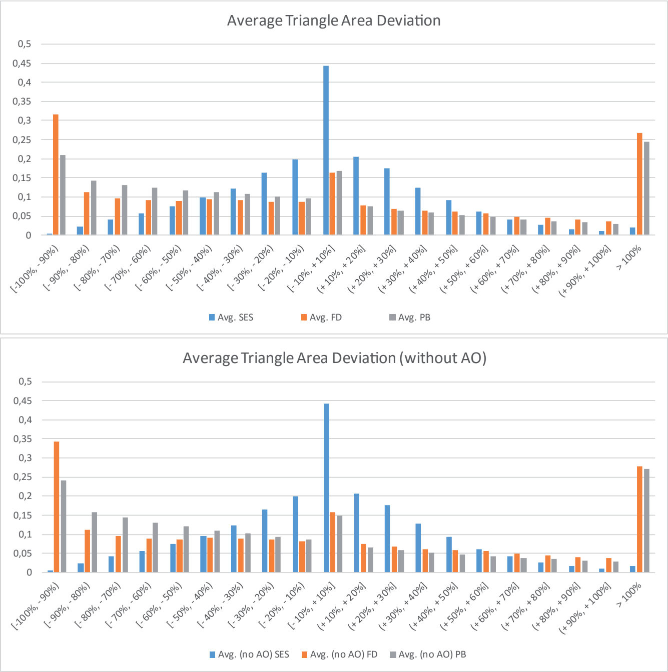

Fig. 12. Histograms showing the average distortion of the triangle area after mapping the SES to the sphere. The deviation was computed with respect to the average triangle size.

As mentioned in Section 3, mapping the SES mesh to the Molecular Surface Globe will result in a non-uniform distribution of triangles. We investigated the distortion created by the mapping based on the resulting triangle areas (see Figure 12). Triangles that lie on protrusions or cavities of the SES will lead to denser regions on the sphere: The triangle area will be lower, which results in a locally higher vertex density. Due to the corrugated nature of the SES, this issue occurs for all geometric mappings that preserve the triangle connectivity. In histograms in Figure 12 show that the parameter-based method creates on average more smaller triangles than the force-directed one. As mentioned in Section 3 turning off the AO for molecules with longer tunnels leads to more distortion (i.e., more smaller triangles, cf. lower histogram in Figure 12). We would rate the parameter-based approach as better suited for more complex surfaces, while the force-directed approach is better suited for more globular molecules without deep cavities or large protrusions.

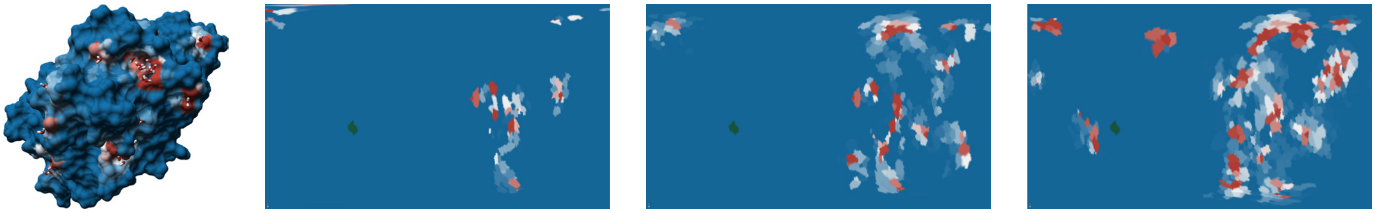

Fig. 13. Left: Solvent Excluded Surface of *Candida antarctica* lipase B (CALB) with water molecules bound to the protein. The surface is colored according to the accumulated water binding over the whole simulation time (blue: low water affinity; red: high water affinity). The three maps to the right show different simulations with increasing water activity, that is, more water molecules (from left to right: 6, 25, and 50 water molecules). The dark green spot shows the location of the ligand binding site. The maps were generated by the force-directed method and Gall-Peters projection

As explained above, the optional AO-based method [29] that removes long channels helps to reduce the distortion. For longer channels, it offers a better placement of cuts than the loop detected using the Reeb graph [13]. For example, without removing long channels from protein 2BT9 shown in Figure 7, the average triangle area decreases from 0.352 to 0.337, while the number of triangles smaller than the average increases from 6573 to 7710. Most importantly, there are now 253 triangles with a size of zero on the map. This means their information is lost because of the distortion.

6.1 Use Cases and Discussion

In this section, we describe common use cases in biochemistry and structural biology that benefit from our Molecular Surface Maps. All use cases were motivated by domain experts and show real-world data. Molecular Surface Maps provide a convenient way to compare multiple molecular data sets. Differences can be spotted easily when viewing two or more maps arranged as small multiples [57] or overlaid without the need for camera adjustment since they are view-independent. In this section, we discuss typical use cases of our project partners from biochemistry were Molecular Surface Maps are applicable. The research of our project partners mostly focuses on lipases. Lipases are often used in industrial biotechnology for synthesis in non-aqueous solvents [50]. Although lipases have only low sequence similarity, they all share the α/β hydrolase fold [37] and a substrate binding site including a catalytic triad consisting of serine, histidine, and aspartic acid. For example, *Candida antarctica* lipase B (CALB) accepts a broad range of substrates [12] and is active at high temperatures [1]. CALB is a widely used lipase both in industrial applications and research, and has been thoroughly studied. However, this enzyme is deactivated by moderate concentrations of short chain alcohols like methanol. Although the inactivation of CALB by methanol has been studied both in experiment and in model [31], there are still several unanswered questions.

Analysis of time-varying data We computed a Space-time Cube of a molecular dynamics simulation of CALB (see Figure 1). The goal of this simulation was to analyze the accessibility of a specific binding site under certain boundary conditions. Therefore, the binding site coloring scheme was chosen. Using classical molecular visualization tools that can only show the SES, the investigation whether the binding site is available at the surface is a time-consuming process, which requires the analyst to inspect each frame of the simulation. Using the Space-time Cube of Molecular Surface Maps, the analyst can immediately see that the red binding site is only reliably accessible for a certain time (see Figure 5). Towards the end of the simulation, it is not accessible at the surface most of the time.

Water binding One interesting and still unsolved question concerning CALB is the role of water. Zaks and Klibanov [62] showed that a small amount of water is necessary for protein activity. Therefore, water can be considered a lubricant that is necessary for enzyme activity. To investigate this, we performed several molecular dynamics simulations of CALB at various water activities, using the GROMACS simulation package and the OPLS force field.

First, our application was used to analyze the distribution of water molecules on the protein surface of a simulation system with a low water activity of 0.1. The coloring mode that shows the aggregated solvent binding for the whole trajectory was used (see Section 4.1). This led to an interesting discovery: Water is not homogeneously distributed over the protein surface, but is binding to several well-defined water binding sites. Molecular Surface Maps quickly revealed that almost all water molecules are bound to one side of the protein, while the other side of the protein is almost completely dry. In contrast to the expectations of our project partners, the binding site of the protein and its active site are almost completely dry.

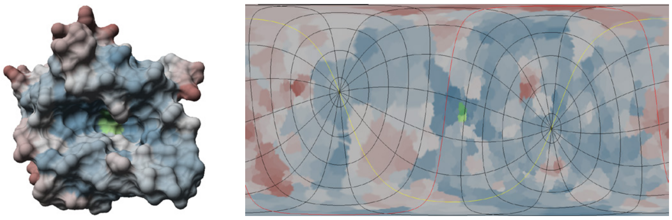

Fig. 14. Analysis of a lipase data set (PDB ID: 1EX9) using Molecular Surface Maps. Two coloring modes are overlaid on the SES (left) and the map (right): color according to the temperature factor (blue-red gradient) and coloring by binding site (green spot). As clearly observable on the map, the green binding site is located in a stable area of the protein (blue). The map was created using the parameter-based method and Hobo-Dyer projection.

The effect of increasing the water activity (by adding more water molecules into the system) was also studied. Using the Molecular Surface Map application, one map per simulated system was produced. The biochemistry experts were pleased with the possibility to easily compare the simulated systems and found interesting results: As expected, with increasing water activity in the simulation system, the amount of water molecules on the protein surface increases. Most of the added water molecules still bind to the already identified high affinity water binding sites. However, additional patches on the protein surface were identified to which water binds only at very high water activities (see Figure 13). These new results can help to better understand the molecular mechanism of enzymes, which can be used to develop strategies to improve the efficiency of proteins in the lab.

Comparison of similar proteins As mentioned earlier, although lipases share a common fold, there are considerable differences between the various lipases. As an initial test, five different lipases (PDB IDs: 1CRL, 1EX9, 1TCA, 1THG, 3TGL) were investigated using Molecular Surface Maps. The summarization provided by the Molecular Surface Map allows for a fast comparison of the different data sets. All lipases catalyze the same type of reactions, however, there are differences in their optimal temperature and their substrate preference. A comparison of both the hydrophobicity and the temperature factor of the abovementioned lipases showed that their surface properties differ largely from each other. While the protein surface of most lipases is divided in lots of small hydrophilic and hydrophobic patches, the surfaces of 1EX9 and 1CRL present larger homogeneous patches. Investigation of the temperature factor, which is an indicator

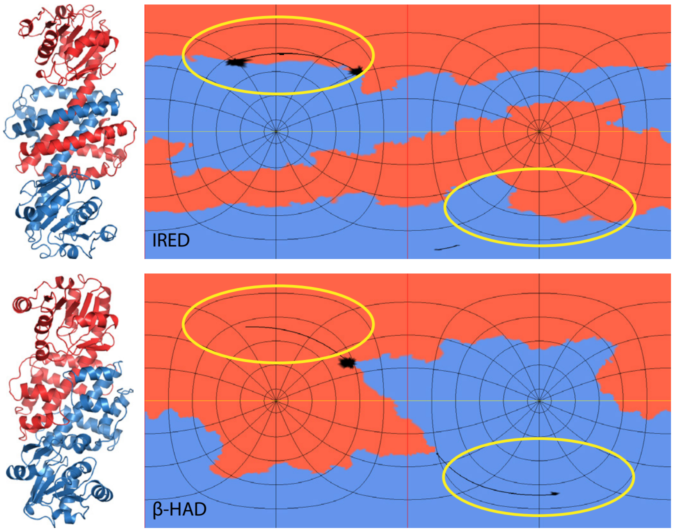

Fig. 15. Comparison of the architecture of IRED (top, PDB ID: 3ZHB) and β-HAD (bottom, PDB ID: 2CVZ). The substrate binding sites are marked by yellow circles on the Molecular Surface Maps, which were created using the parameter-based method and Plate Carrée projection.

for the stability of the protein, revealed that all lipases are relatively stable with the exception of 1EX9, which has many large, flexible regions. The binding site coloring of the maps can reveal whether a lipase is in the open state (i.e., the binding site is accessible at the surface) or in the closed state (binding site inaccessible). Identifying and understanding such differences in the protein interface is crucial for understanding and predicting the effect of mutations. With our Molecular Surface Map application, users get an overview of these surface properties. Our project partners especially liked the possibility to overlay two maps of the same protein, since this can also reveal interesting phenomena. This for example showed that although lipase 1EX9 is overall much more flexible than the other four lipases, the binding site is not only accessible but also lies in a very stable region (see Figure 14). Comparing maps of different proteins side by side allows users to assess their differences in a quick and convenient way compared to conventional molecular visualization tools.

Investigation of protein architecture Imine reductases (IREDs) and β-hydroxyacid dehydrogenases (β-HADs) catalyze the stereoselective reduction of their substrates, imines or alcohols, respectively, to industrial and pharmaceutically relevant products [49, 63]. Both enzymes form homodimers, a complex of two identical proteins. Despite the differences in the substrate specificity, both families are similar on sequence level and show a similar overall structure [15]. However, due to a split helix in the β-HAD, dimer arrangement differs in the two enzymes (see Figure 15, left). To visualize the differences, Molecular Surface Maps were applied and demonstrated clearly that the substrate binding sites are formed differently: As observable in Figure 15, the binding site of β-HAD is build only by one monomer, while both monomers contribute to the binding site in IRED. The domain experts were very pleased with Molecular Surface Maps for this task, since they show the structural differences in a clear and comprehensive way.

In summary, the experts from the field of biochemistry rated Molecular Surface Maps as a useful tool for several applications in bioinformatics. They believe that they can be applied also to other enzyme families, thus helping to gain further insights, and rated them as a useful addition to established molecular visualizations.

7 Summary and Future Work

We presented the Molecular Surface Map application that provides a novel and concise way of visualizing molecular surfaces and their respective attributes in a planar representation. We achieve this by using a morphing algorithm that maps the original surface to a sphere. If the molecular surface has a genus greater than zero, all holes are closed prior to the morphing. Afterwards this sphere—the Molecular Surface Globe—is transferred to planar domain using a map projection.

We applied our application to several real world data sets and, thereby, showed how our Molecular Surface Maps application can be used to analyze different attributes, such as the location of binding sites or biochemical properties. We extended our visualization to dynamic data using either aggregation or a Space-time Cube, which allows to track important molecular surface features over time. Our application was developed in close collaboration with domain scientists from biochemistry to ensure that it is feasible for their use cases.

In the future, we want to improve the visual coherence between the Molecular Surface Maps and the original surface. One way could be to draw contour lines based on the elevation that highlight the locations of cavities or protrusions and their depth or height, respectively. Furthermore, we want to offer more map projection schemes, for example conformal projection like Peirce quincuncial or elliptical projections like Mollweide, which have lower distortion at the poles [53]. The Myriahedral projections introduced by van Wijk [58] could also be used if low distortion is desired. We plan to evaluate the feasibility of different map projections together with domain scientist. Another possible enrichment of the map would be to use icons or text labels to depict important features. This could also be used for user annotations.

Furthermore, we also to investigate the feasibility of further Space-time Cube operations, for example *space flattening*, which can be used to reduce the dimensionality. Similar Space-time Cube visualizations have for example been applied successfully for the visual analysis of eye tracking data [32]. This would lead to an even more concise representation of the development of the visualized molecular surface features (e.g., the temporary accessibility of binding sites).

The removal of channels of the molecule can lead to better mappings, however, the information within the channel will be lost. In the future, we want to extend our Molecular Surface Map application to visualize these removed inner structures separately, especially for complex channel networks. One possibility would be a planar visualization that maintains the spatial relations, similar to the one for vessels trees that was recently presented by Marino and Kaufman [36].

Acknowledgments

The authors want to thank M. Sanner for providing his *MSMS* software and T. K. Dey and colleagues for providing their *ReebHanTun* software. This work was partially funded by Deutsche Forschungsgemeinschaft (DFG) as part of SFB 716 and SFB-TRR 161.

References

[1] E. M. Anderson, K. M. Larsson, and O. Kirk. One biocatalyst–many applications: the use of candida antarctica b-lipase in organic synthesis. *Biocatal. Biotransfor.*, 16(3):181–204, 1998.

[2] S. Anzali, G. Barnickel, M. Krug, J. Sadowski, M. Wagener, J. Gasteiger, and J. Polanski. The comparison of geometric and electronic properties of molecular surfaces by neural networks. *J. Comput. Mol. Des.*, 10(6):521–534, 1996.

[3] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A Review of Temporal Data Visualizations Based on Space-Time Cube Operations. In *EuroVis - STARs*, volume 1, pages 23–41, 2014.

[4] D. Baum and H.-C. Hege. A point-matching based algorithm for 3d surface alignment of drug-sized molecules. In *Computational Life Sciences II*, volume 4216 of *Lecture Notes in Comput. Sci.*, pages 183–193. 2006.

[5] J. Bennett, V. Pascucci, and K. Joy. Genus Oblivious Cross Parameterization: Robust Topological Management of Inter-Surface Maps. In *PacificGraphics*, pages 238–247, 2007.

[6] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000. `http://www.pdb.org`.

[7] J. F. Blinn. A Generalization of Algebraic Surface Drawing. *ACM TOG*, 1(3):235–256, 1982.

[8] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[9] D. Borland. Ambient Occlusion Opacity Mapping for Visualization of Internal Molecular Structure. *Journal of WSCG*, 19(1):17–24, 2011.

[10] M. L. Connolly. Analytical Molecular Surface Calculation. *J. Appl. Cryst.*, 16(5):548–558, 1983.

[11] D. Cosgrove, D. Bayada, and A. Johnson. A novel method of aligning molecules by local surface shape similarity. *J. Comput. Aided Mol. Des.*, 14(6):573–591, 2000.

[12] P. Degn and W. Zimmermann. Optimization of carbohydrate fatty acid ester synthesis in organic media by a lipase from candida antarctica. *Biotechnol. Bioeng.*, 74(6):483–491, 2001.

[13] T. K. Dey, F. Fan, and Y. Wang. An Efficient Computation of Handle and Tunnel Loops via Reeb Graphs. *ACM Transactions on Graphics*, 32(4):32:1–32:10, 2013.

[14] H. Edelsbrunner. Deformable Smooth Surface Design. *Discrete Comput. Geom.*, 21(1):87–115, 1999.

[15] S. Fademrecht, P. N. Scheller, B. M. Nestl, B. Hauer, and J. Pleiss. Identification of imine reductase-specific sequence motifs. *Proteins*, 2016.

[16] C. Faloutsos and K. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *ACM SIGMOD*, pages 163–174, 1995.

[17] X. Gu and S.-T. Yau. Global Conformal Surface Parameterization. In *EG/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 127–137, 2003.

[18] T. Hägerstrand. What about people in Regional Science? *Pap. Reg. Sci.*, 24(1):6–21, 1970.

[19] K. Hasegawa and K. Funatsu. New description of protein-ligand interactions using a spherical self-organizing map. *Bioorganic & Medicinal Chemistry*, 20(18):5410–5415, 2012.

[20] J. Hass and P. Koehl. How round is a protein? Exploring protein structures for globularity using conformal mapping. *Mathematics of Biomolecules*, 1:26, 2014.

[21] A. Henderson, J. Ahrens, and C. Law. The ParaView Guide, 2004.

[22] C. Hofbauer, H. Lohninger, and A. Aszódi. SURFCOMP: A novel graph-based approach to molecular surface comparison. *J. Chem. Inf. Comput. Sci*, 44:837–847, 2004.

[23] K. Hormann, B. Lvy, and A. Sheffer. Mesh Parameterization. In *ACM SIGGRAPH Courses*, 2007.

[24] D. G. Kontopoulos, D. Vlachakis, G. Tsiliki, and S. Kossida. Structuprint: a scalable and extensible tool for two-dimensional representation of protein surfaces. *BMC Struct. Biol.*, 16:4, 2016.

[25] B. Kozlikova, M. Krone, N. Lindow, M. Falk, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of Biomolecular Structures: State of the Art. In *Eurographics Conference on Visualization - STARs*, pages 61–81, 2015.

[26] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3d models. In *ACM Transactions on Graphics*, volume 23, pages 861–869, 2004.

[27] M. Krone, S. Grottel, and T. Ertl. Parallel Contour-Buildup Algorithm for the Molecular Surface. In *IEEE Symposium on Biological Data Visualization*, pages 17–22, 2011.

[28] M. Krone, D. Kauker, G. Reina, and T. Ertl. Visual Analysis of Dynamic Protein Cavities and Binding Sites. In *IEEE PacificVis - Visualization Notes*, volume 1, pages 301–305, 2014.

[29] M. Krone, G. Reina, C. Schulz, T. Kulschewski, J. Pleiss, and T. Ertl. Interactive Extraction and Tracking of Biomolecular Surface Features. *Comput. Graph. Forum*, 32(3):331–340, 2013.

[30] J. Krüger and R. Westermann. Acceleration Techniques for GPU-based Volume Rendering. In *IEEE Visualization*, pages 287–292, 2003.

[31] T. Kulschewski, F. Sasso, F. Secundo, M. Lotti, and J. Pleiss. Molecular mechanism of deactivation of c. antarctica lipase b by methanol. *J. Biotechnol.*, 168(4):462–469, 2013.

[32] K. Kurzhals and D. Weiskopf. Space-Time Visual Analytics of Eye-Tracking Data for Dynamic Stimuli. *IEEE Trans. Vis. Comput. Graphics*, 19(12):2129–2138, 2013.

[33] D. La, J. Esquivel-Rodríguez, V. Venkatraman, B. Li, L. Sael, S. Ueng, S. Ahrendt, and D. Kihara. 3D-Surfer: Software for High-throughput Protein Surface Comparison and Analysis. *Bioinformatics*, 25(21):2843–4, 2009.

[34] N. Lindow, D. Baum, S. Prohaska, and H.-C. Hege. Accelerated Visualization of Dynamic Molecular Surfaces. *Comput. Graph. Forum*, 29(3):943–952, 2010.

[35] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *Proc. 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 27–34, 1993.

[36] J. Marino and A. Kaufman. Planar Visualization of Treelike Structures. *IEEE Trans. Vis. Comput. Graphics*, 22(1):906–915, 2016.

[37] M. Nardini and B. W. Dijkstra. α/β hydrolase fold enzymes: the family keeps growing. *Curr. Opin. Struc. Biol.*, 9(6):732–737, 1999.

[38] A. A. Polyansky, A. O. Chugunov, P. E. Volynsky, N. A. Krylov, D. E. Nolde, and R. G. Efremov. PREDDIMER: a web server for prediction of transmembrane helical dimers. *Bioinformatics*, page btt645, 2013.

[39] N. Postarnakevich and R. Singh. Global-to-local representation and visualization of molecular surfaces using deformable models. In *ACM Symp. Appl. Comput.*, pages 782–787, 2009.

[40] E. Praun and H. Hoppe. Spherical parametrization and remeshing. In *ACM TOG*, volume 22, pages 340–349, 2003.

[41] S. J. Rahi and K. Sharp. Mapping complicated surfaces onto a sphere. *Int. J. Comput. Geom. Appl.*, 17(04):305–329, 2007.

[42] G. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.*, 7(1):95–99, 1963.

[43] F. M. Richards. Areas, Volumes, Packing, and Protein Structure. *Annual Review of Biophysics and Bioengineering*, 6(1):151–176, 1977.

[44] J. S. Richardson. The Anatomy and Taxonomy of Protein Structure. *Adv. Protein Chem.*, 34:167–339, 1981.

[45] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[46] S. Saba, I. Yavneh, C. Gotsman, and A. Sheffer. Practical spherical embedding of manifold triangle meshes. In *Intl. Conf. on Shape Modeling and Applications*, pages 256–265, 2005.

[47] M. F. Sanner, A. J. Olson, and J.-C. Spehner. Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers*, 38(3):305–320, 1996.

[48] K. Scharnowski, M. Krone, G. Reina, T. Kulschewski, J. Pleiss, and T. Ertl. Comparative visualization of molecular surfaces using deformable models. *Comput. Graph. Forum*, 33(3):191–200, 2014.

[49] P. N. Scheller, S. Fademrecht, S. Hofelzer, J. Pleiss, F. Leipold, N. J. Turner, B. M. Nestl, and B. Hauer. Enzyme Toolbox: Novel Enantiocomplementary Imine Reductases. *Chem. Bio. Chem.*, 15:2201–2204, 2014.

[50] R. D. Schmid and R. Verger. Lipases: interfacial enzymes with attractive applications. *Angew. Chem. Int. Edit.*, 37(12):1608–1633, 1998.

[51] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-Surface Mapping. *ACM TOG*, 23(3):870–877, 2004.

[52] T. Shen, X. Huang, H. Li, E. Kim, S. Zhang, and J. Huang. A 3D Laplacian-driven parametric deformable model. In *ICCV*, pages 279–286, 2011.

[53] J. P. Snyder and P. M. Voxland. An Album of Map Projections. USGS Numbered Series 1453, 1989.

[54] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable Medical Image Registration: A Survey. *IEEE Trans. Med. Imaging*, 32(7):1153–1190, 2013.

[55] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: recovering 3D shape and nongrid motion. *Artif. Intell.*, 36(1):91–123, 1988.

[56] B. Tian. *Bouncing Bubble: A fast algorithm for Minimal Enclosing Ball problem*. GRIN Publishing, 2012.

[57] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition edition, 2001.

[58] J. J. van Wijk. Unfolding the Earth: Myriahedral Projections. *The Cartographic Journal*, 45(1):32–42, 2008.

[59] M.-Q. Wei, M.-Y. Pang, and C.-L. Fan. Survey on Planar Parameterization of Triangular Meshes. In *2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, volume 3, pages 702–705, 2010.

[60] M. Weisel, E. Proschak, and G. Schneider. PocketPicker: Analysis of ligand binding-sites with shape descriptors. *Chem. Cent. J.*, 1(1):7, 2007.

[61] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *Image Process. IEEE Trans.*, 7(3):359–369, 1998.

[62] A. Zaks and A. M. Klibanov. The effect of water on enzyme action in organic media. *J. Biol. Chem.*, 263(17):8017–8021, 1988.

[63] Y. Zhang, Y. Zheng, L. Qin, S. Wang, G. W. Buchko, and R. M. Garavito. Structural characterization of a β-hydroxyacid dehydrogenase from Geobacter sulfurreducens and Geobacter metallireducens with succinic semialdehyde reductase activity. *Biochimie*, 104:61–69, 2014.

[64] S. Zhukov, A. Iones, and G. Kronin. An Ambient Light Illumination Model. In *Eurographics Workshop on Rendering*, pages 45–56, 1998.