








Visual Analysis of Large-Scale Protein-Ligand Interaction Data

Karsten Schatz,¹  Juan José Franco-Moreno,²  Marco Schäfer,³  Alexander S. Rose,⁴  Valerio Ferrario,⁵  Jürgen Pleiss,⁵ 
Pere-Pau Vázquez,²  Thomas Ertl¹  and Michael Krone³ 

¹Visualization Research Center (VISUS), University of Stuttgart, Stuttgart, Germany
{karsten.schatz, thomas.ertl}@visus.uni-stuttgart.de

²ViRVIG Group, UPC Barcelona, Barcelona, Spain
{juan.jose.franco.moreno, pere.pau.vazquez}@upc.edu

³Big Data Visual Analytics in Life Sciences (BDVA), University of Tübingen, Tübingen, Germany
{marco.schaefer, michael.krone}@uni-tuebingen.de

⁴RCSB Protein Data Bank, San Diego Supercomputer Center, University of California, La Jolla, San Diego, California, USA
alex.rose@rcsb.org

⁵Institute of Biochemistry and Technical Biochemistry, University of Stuttgart, Stuttgart, Germany
{valerio.ferrario, juergen.pleiss}@itb.uni-stuttgart.de

Abstract

When studying protein-ligand interactions, many different factors can influence the behaviour of the protein as well as the ligands. Molecular visualisation tools typically concentrate on the movement of single ligand molecules; however, viewing only one molecule can merely provide a hint of the overall behaviour of the system. To tackle this issue, we do not focus on the visualisation of the local actions of individual ligand molecules but on the influence of a protein and their overall movement. Since the simulations required to study these problems can have millions of time steps, our presented system decouples visualisation and data preprocessing: our preprocessing pipeline aggregates the movement of ligand molecules relative to a receptor protein. For data analysis, we present a web-based visualisation application that combines multiple linked 2D and 3D views that display the previously calculated data. The central view, a novel enhanced sequence diagram that shows the calculated values, is linked to a traditional surface visualisation of the protein. This results in an interactive visualisation that is independent of the size of the underlying data, since the memory footprint of the aggregated data for visualisation is constant and very low, even if the raw input consisted of several terabytes.

Keywords: scientific visualisation, visualisation, protein-ligand interaction

ACM CCS: • Human-centred computing → Visualization systems and tools; • Applied computing → Molecular structural biology

1. Introduction

Protein-ligand interactions are a vast and diverse field. Different techniques and approaches have been developed so far for understanding the complex and specific interactions between ligand molecules and protein surfaces. Simulations that are performed to study this kind of interactions tend to comprise terabytes of data or even more. While existing approaches for the visualisation of protein-ligand interactions, like the one by Vázquez et al. [VHG*18], typically concentrate on single ligand molecules, global approaches to analyse ligand movement are rare. As a direct visualisation of the complete data sets is normally impossible

due to their size, approaches that employ streaming or aggregation of the data are the most viable options. Streaming approaches typically use animation and are often preferred by domain scientists as they are easy to understand. However, since it is well known that the analysis of animations, especially when they are long, is more difficult to humans than the analysis of static images [TMB02], an aggregated representation of the simulations can be more beneficial for an adequate visualisation. Unfortunately, systems that use aggregated data keep track of a limited number of represented parameters. Moreover, they do not calculate information such as contact counts, that are highly relevant for our collaborators.

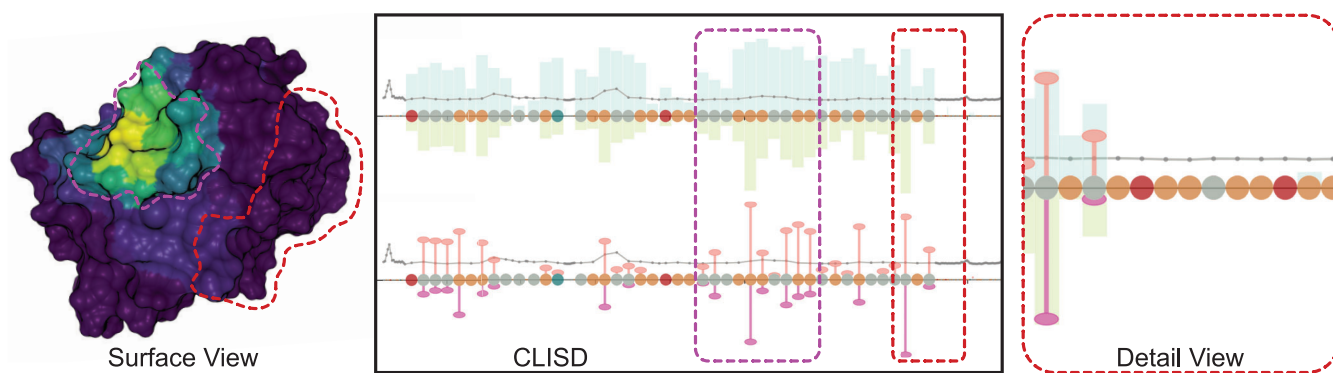



Figure 1: Information shown in the three main views of our protein-ligand interaction visualisation: The x-axes denote the amino acids of the protein. The bars above the x-axes show the number of time steps near a ligand molecule, the bars below show the number of close ligand atoms. The width of the bars in the stackable CLISD (Compressed Ligand Interaction Sequence Diagram) is scaled according to the value, thus emphasizing higher, more important values. De-emphasised values can be inspected in the detail view, alongside additional bond counts. The surface view can be coloured by different aggregated quantities using the Viridis colour map (; purple corresponds to 0 and yellow to the maximum value). Here, the amount of time steps close to a ligand atom is depicted (lower green bars of the diagram).

Our collaborators want to understand how and where ligand molecules approach the active site of an enzyme. To do so, they generate extremely large simulations that cannot be tackled by almost any existing software directly. For example, packages such as MegaMol [GKM*15] would be able to reproduce the animation, but do not provide tools for the analysis. Other packages with a scriptable interface would of course support any dataset indirectly, but they require upfront work by the user and visual programming expertise which cannot be expected from domain scientists. To effectively inspect those simulations, domain experts are interested in the interactions/contacts between the enzyme and the ligand. Therefore, in collaboration with the co-authoring domain experts, we formulated seven design requirements for our software that tackles this problem. We have developed a system that facilitates the visual analysis of protein-ligand interactions from whole Molecular Dynamics (MD) simulation trajectory ensembles. Prior to the visualisation, we aggregate the relevant values of the simulation data. This way, we can provide a meaningful overview of the whole simulation that allows scientists to further investigate interesting steps. For the subsequent data analysis, we developed a web-based multi-view visualisation application that is specifically designed to show these aggregated results. We introduce the Compressed Ligand-Interaction Sequence Diagram (CLISD) that provides an overview of the amino acids of the protein that interact with the ligands (cf. Figure 1). Our sequence diagram is enhanced by the aggregated values and is designed to draw the attention of the user to the relevant parts. The two-dimensional CLISD is linked to a traditional, three-dimensional surface visualisation of the protein, which can be coloured according to the aggregated values. This molecular surface visualisation enables users to analyse the spatial structure of the depicted molecule.

Our contributions can be summarised as follows: We propose a preprocessing pipeline for several specific values to aggregate and derive the movement of ligand molecules relative to a receptor protein (e.g. during a simulation) and present a web-based visualisa-

tion application that consists of multiple linked 2D and 3D views that facilitate the visual analysis of the previously aggregated data. Specifically, we introduce the CLISD that shows which amino acids of the protein interact with the ligand. Our interactive visualisation is specifically designed for the exploratory analysis of protein-ligand interactions from MD simulation data, but it can also be used to analyse docking results and the interaction of the protein with substrate and solvent molecules. We believe our system could additionally be used to improve biochemical properties of the enzymes such as substrate affinity and catalytic activity by targeted mutation of amino acids. We demonstrate the capabilities of our approach in two use cases that were investigated in tight collaboration with our project partners from biology.

2. Structural Biology Background

This section introduces the biological background of our work that targets the interaction of proteins with small molecules. Proteins are biological macromolecules that consist of one or more chains of amino acids (or *residues*). The folding of these chains into the energetically most stable conformation determines the spatial structure of the protein.

Proteins serve many tasks; may it be in the bodies of living beings or for biotechnological applications like biofuel production. The most important proteins for our work are enzymes, which can trigger or accelerate chemical reactions of other smaller molecules. Molecules that are able to specifically bind to the proteins are called ligands, or more enzyme-specific, substrate molecules. Ligands may alter the behaviour of the protein or undergo a chemical reaction when reaching a specific part of the protein called the active site. The surrounding surface geometry and dynamics of the protein influence the types of ligands that can reach that active site. Often, only few selected ligands are actually able to reach the active site. In literature, this specificity for docking ligands is often described

as lock-and-key model [Fis94]. Another important factor, especially for enzymes, is the reaction rate of the protein-ligand system. Surface properties of the protein as well as the ligand geometry may influence this rate.

One major field of research is the engineering of enzymes to increase their reaction rates. To reach this goal, specific amino acids of the protein chains are inserted, deleted, or replaced by other ones. This may alter the behaviour of the protein as these modifications can change the stability or influence the accessibility to the active site. Typically, a simulation of the altered protein is performed to assess changes in its behaviour (*in silico*). Real-life experiments (*in vivo*) with artificially produced proteins are ideally only carried out if the simulation results were positive. The goal of our work is not only to ease the analysis of the simulation results but also to provide a tool able to support the work of protein engineers by making the understanding of simulations more clear and accessible. Apart from the physico-chemical surface properties, the access of a ligand to the active site is, to a significant amount, a geometrical problem. Thus, our tool supports the visual inspection of protein geometry.

Since ligand interactions happen at the protein surface, good surface representations are necessary. The most commonly used and simple molecular surface is the van der Waals (vdW) surface [vdW73]. It represents each atom by a sphere of fixed radius that depends on the element of the atom. Based on this method, two closely related molecular surfaces can be derived: the Solvent Accessible Surface (SAS) [LR71] and the Solvent Excluded Surface (SES) [Con83, Ric77]. Both are defined by rolling a spherical probe around the vdW surface. The radius of the probe sphere corresponds to the size of the assumed ligand. The area described by the centre of the probe is called the SAS, the area the probe is not able to reach without intersecting the vdW surface is called the SES. A survey of molecular surfaces was recently given by Kozlíková et al. [KKF*16]. We use the SES in our work since it does not hide a ligand docked to the surface.

3. Related Work

Since its inception several decades ago, visualisation techniques purposely designed for biomolecular structures have played a key role in the understanding and discovery of chemical phenomena. The extension and depth of this research area is illustrated by the number of recent surveys [KKF*16, KKL*16, AAM*17, SKPE19] that deal with different subtopics. The improvements in simulation algorithms and computational power continuously challenge previously developed visualisation techniques. On the one hand, the models are increasing in size and complexity. The number of elements—for example atoms or cells—to represent visually surpassed the order of millions quite some time ago [GKM*15, LMAPV15]. On the other hand, the (non-geometric) information researchers need to explore is also continuously increasing, for example, the ever growing time step count in molecular simulations [DHR*19], or the large number of properties to represent (such as water trajectories [VBJ*17], factorised energy components [VHG*18] or the chemical properties and interactions within protein cavities [FJB*17, BLMG*16, BJB*15]). Hence, on top of the geometric complexity, it is necessary to find room and compu-

tational power to deal with such complex data sets in order to make them useful and understandable for researchers.

Traditional visualisation research often focuses on using GPUs to accelerate the rendering of complex 3D structures using different strategies, such as Level-of-Detail [IWR*18] or highly parallel architectures [RHI*15, KWN*14]. However, if we analyse recent visualisation approaches that address the data complexity problem, smart data abstractions and aggregation techniques to explain more in smaller space can be found increasingly. Thus, we focus on two different areas: the aggregation of the massive input data to extract meaningful information, and the design of a multi-view visualisation system that enables the exploration of a large number of variables at once.

Data Aggregation. Visual analysis is at the core of all visualisation techniques [HS12]. When the data becomes very large, for example when visualising whole ensembles, several strategies can be employed to reduce the required space. Animation, for instance, reuses the same space by modifying the visual depictions with the time. However, for large sequences, animation can be less effective than static graphs [TMB02], as it relies on the short-term memory of viewer [WHL19]. The opposite approach is to aggregate the information so that it can be digested and facilitate detailed exploration of elements of interest. In molecular visualisation, *abstraction* and level-of-detail techniques have been used extensively (e.g. [MDLI*18]), since they facilitate the reduction of clutter when the size or density of elements requires a sizable screen footprint. Other methods have been developed to represent full molecules [PRV13, PJR*14], as well as other information such as solvent pathlines near protein cavities [BGB*08]. *Data aggregation* has also been used for non-geometric properties, such as energy plots. For instance, Duran et al. [DHR*19] presented simplified energy charts using a hierarchical exploration tool that supports the exploration of detailed parts with few clicks. Data aggregation can be very useful if it adequately provides an informative overview of the whole data set. Vázquez et al. [VHG*18] aggregate energy data from a whole Monte Carlo-based molecular simulation to facilitate the quick exploration of interaction areas. However, their approach deals with relatively short trajectories (several hundreds of steps) and does not include other interesting information such as contacts. Bidmon et al. [BGB*08] cluster the paths of solvent molecules near the active site of a protein to simplify the exploration of complex solvent movement patterns. Skånberg et al. [SKL*18], on the contrary, facilitate the exploration of large MD data with an extensive use of semantic *filtering* operations. Byška et al. [BTM*19] detect interesting spatio-temporal events in very large simulations so that the users can concentrate on these potentially interesting parts and filter out uninteresting portions of the simulations. Alharbi et al. [ALC16] facilitate the exploration of MD simulations by path filtering based on geometric properties of the paths, such as edge lengths or curvatures. In a subsequent work [AKCL19], they presented a tool that extracts and visualises protein-protein interactions as well as protein-lipid ones from MD simulations. However, their method is tailored to membrane simulations and cannot be applied directly to protein-ligand interactions.

Visualisation of Multiple Variables. Most visualisation systems use multi-view and overlaying techniques to increase the number of variables that are shown at once. In molecular visualisation, for

example, Vázquez et al. [VHG*18] create a compact visualisation for the depiction of protein-ligand binding simulations. They overlay up multiple variables at once: molecular backbone, per-residue energy, minimum and maximum energies, h-bonds, etc. They combine both single step and aggregated information. Hermosilla et al. [HEG*17] display multiple energy-related information and make extensive use of filtering operations to display the interactions between the ligands and the proteins in a simulation on a step-by-step basis. However, this method does not give an overview of the whole simulation. Lichtenberg et al. [LMA*18] propose the Residue Surface Proximity map, which allows users to visually analyse the closeness of each amino acid of a protein to the surface throughout a simulation. This corresponds to the potential exposure to ligands, it is contrasting to our approach, which focuses on the actual behaviour of the ligand molecules. Furmanová et al. [FJK*19] facilitate the exploration of protein-protein contacts through a combination of charts and a 3D view. However, they concentrate on the analysis of contacts and provide views for illustrating the individual residues' interaction in a node-link view. In our case, we are more interested in showing the whole path in an exploratory way and providing information such as the number of steps in contact. Mostly, we define a contact as spatial proximity without the forming of covalent bonds.

Our strategy for coping with large amounts of data, we use aggregation. As a result, we are able to generate informative overviews of the whole simulation, and the resulting visualisation tool can also run on commodity hardware. Exploratory visualisation is achieved through data superposition, filtering, and compaction to maximise the amount of information that can be displayed at the same time.

4. Design Requirements

Our goal was to create a visual analysis application for protein-ligand interactions that is able to visualise even the largest biomolecular data sets (e.g. ensembles of MD simulation data). In order to be effectively usable by biologists and to integrate smoothly into their everyday analysis workflow, it is important to listen to the needs of the actual target users. In close cooperation after numerous refining iterations, we identified seven requirements that shall be fulfilled by the resulting application.

The first requirement R1 stems from the sheer size of the data our collaboration partners are working with. Although they are typically used to animations, we quickly realised that this is nearly impossible for the current data at hand. Even with filtering approaches, tens of thousands of time steps would remain relevant. Animations of solely those parts alone move too fast for a human to enable the observer to analyse the presented data in full detail without jumping back and forth, especially when also the subsequent requirements have to be considered. In aggregated visualisations always some part of the original information is lost. While the following four requirements mostly specify what our domain scientist would like to see in a visualisation, they also restrict which information we are allowed to omit in the aggregation process. R2 therefore enforces the inclusion of the location, duration, and frequency of contacts between residues of the proteins and ligand molecules. As the sequence, as well as the folding of its underlying amino acid chain, influences

the behaviour of a protein, the requirement also enforces the usage of certain visual representations.

The most interesting part for our domain scientists were the actual paths of the ligands on the surface of the protein, as all other tools they knew either did not work or produced too much visual clutter. To investigate the hypothesis that ligands crawl along the surface, the landing spots were of keen interest to them, followed by the actual path taken (R5 and R3). However, as different kinds of ligands form different kinds of bonds, R4 becomes important. Most ligand types are able to form hydrogen bonds, for example, but other contact definitions are possible. Although pure distance-based contact definitions still stay relevant in the general case, specialised definitions should be allowed and usable.

Two further requirements stem from the workflow of the domain scientists. Although they can obtain some of the analysis data with existing tools such as GROMACS [AMS*15] or the MDAnalysis library [MDWB11, GLB*16] by analyzing distances and interactions from the simulation trajectory, it is currently not possible to extract all of the information presented by us. To the best of our knowledge, all the available tools either require to load the full trajectory into memory, which is only possible on dedicated hardware with high memory, or they require extensive coding efforts. Dedicated solutions to reduce the memory requirement are currently not available in the existing analysis libraries. Furthermore, dedicated tools to aggregate and visualise the analysed data are not available which makes the systematic analysis of large simulation data complex and time consuming.

In addition, a graphical representation of aggregated data is missing in the available tools that are typically tailored towards computing clusters. For visualisation, the domain scientists typically use PyMol [SD] or VMD [HDS96]. While both programs provide many options for surface visualisations, features for sequence diagrams are limited or not present at all. Furthermore, both programs have neither a dedicated way to input aggregated data, nor are they able to depict longer trajectories without loading them completely into RAM. As it is mostly impossible for the domain scientists to occupy a special machine for visualisation tasks, requirement R6 was formulated. This, of course, leads to longer waiting periods for the precalculations. For them, this is no issue, as they are already used to waiting periods for their MD simulations, for example. As long as these calculations are not necessary each time one starts the visualisation application, they are acceptable (R7). To sum up, the seven found requirements are:

R1 The visualisation should give an overview of the whole simulation, instead of showing the whole simulation as an animation.

R2 Areas where contacts are most common should be identifiable in a surface representation as well as in the amino acid sequence.

R3 Ligand paths on the surface of the protein should be identifiable.

R4 Depending on the use case, different kinds of contact definitions need to be considered (e.g. distance-based, or hydrogen bonds).

R5 Identification of areas/amino acids where ligands get initially into contact with the protein (“*landing spots*”).

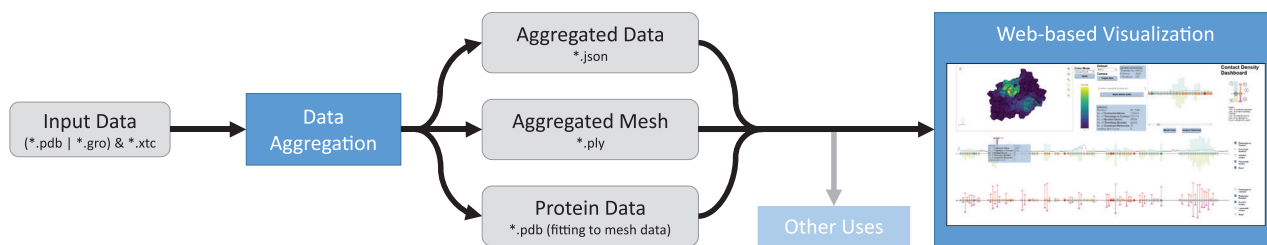


Figure 2: Overview of the data pipeline of our system. First, the input data, given as Protein Data Bank or GROMACS files alongside with a XTC trajectory is analysed and accumulated step by step. After the accumulation is finished, three types of files are written: Human-readable json-files that contain the measured results, a representative surface mesh containing colour values for each of the accumulated quantities, and a Protein Data Bank file providing atom position data fitting to the surface mesh. These files can either be directly shown with our web-based visualisation framework, or further calculations can be performed using other programs. The Polygon File Format meshes, for example, can be read by standard mesh viewers (e.g. MeshLab). The inset on the right is a screenshot of the whole web-based application.

R6 The preprocessing as well as the visualisation should be possible on a commodity machine.

R7 It should be possible to present visualisation results to collaborators without having to wait more than a few seconds.

The first five design requirements formulate requests for the outcome of the visualisation, the last two restrict the technical implementation. Please note that R3 and R5 are not applicable to protein docking data, as they were formulated with the more complex use case of simulation data in mind.

5. Overall Approach and Application Overview

In the light of the self-given requirements, especially R1, we decided to use an aggregation approach, instead of pure filtering (e.g. Alharbi et al. [ALC16], cf. Section 3). Showing such temporally aggregated values in a static visualisation eases the analysis since no animation is required. It also has the benefit that it can reduce the amount of data required for the visualisation tremendously compared to the original simulation data set size, as detailed below. Additionally, it allows us to completely decouple the aggregation in a preprocessing step from the visualisation.

For the actual visualisation, we decided to implement a web-based application since this is most convenient for the domain scientists, as it will be accessed via a browser from all devices without any installation (requirement R6). The final application should primarily incorporate visualisation methods that are familiar to users from the field of biology to further reduce the barriers to usage and facilitate an intuitive and fast analysis process. This includes commonly used three-dimensional visualisations of protein structures as well as abstract, two-dimensional representations, such as sequence diagrams that depict the linear chain of amino acids. Both types of representations can be enhanced by incorporating the values that are relevant for protein-ligand interactions (e.g. by colour-mapping them onto the representation). As the interactions between a protein and ligands highly depend on surface geometry, presenting only an enhanced sequence diagram would not be sufficient. A molecular surface visualisation such as the SES is more suitable to show the spatial relations between molecules and depicts the conformation

of the protein. However, as some amino acids are not part of a protein's surface all the time, a surface visualisation alone is also not sufficient. Furthermore, exact numeric values can be assessed better in a 2D plot than in a 3D visualisation. Therefore, we use both representations by showing the SES of the protein from a representative time step alongside an enhanced sequence diagram. This is also represented by requirement R2. Our web-based visualisation tool thus combines the advantages of both depictions and links both views with suitable interaction techniques. A detailed description of the visualisation capabilities of our application—especially the enhanced sequence diagram—is given in Section 7.

An overview of our proposed pipeline, including both the data processing and the visualisation, is shown in Figure 2. We start with input data coming from MD simulations or docking experiments. In the following, we will refer to the first kind as *simulation data* and for the latter one as *non-simulation data*. Our data processing application loads the data step by step. Only after the current time step is processed, the next one is loaded. As a result, the memory footprint of the preprocessing application always stays constant and does not increase with input simulation length. As demanded by requirement R6, this allows commodity machines to perform the aggregation and especially to store and visualise the results. The main idea of the proposed preprocessing is that the values are temporally aggregated per amino acid so that the output is independent of the number of input time steps. After the accumulation is performed, three kinds of data are available: the aggregated values per amino acid (consisting of aggregated raw and newly derived data), an aggregated SES mesh containing colours for each variable, and the protein data, which contains basic information about the protein structure such as atom types and positions. This information can then be loaded by the web-based visualisation application for an exploratory analysis of the aggregated values. Section 8 describes the implementation details of both applications as well as the file formats chosen for data exchange.

6. Data Processing

The first step of our pipeline is to process and aggregate the input data, which summarises the data and makes it small enough to be stored and transferred to our web-based visualisation described in

Section 8.2. It consists of three sub-steps: first, the data preparation that brings all given time steps into the same reference frame. Second, deriving required quantities from each time step. Third, and finally, aggregating all read and derived data to produce a visualisable result. The main focus lies on single or multiple simulation runs, but we also support ligand docking results. Since simulation trajectories can be very large, most of the properties for aggregation were chosen in a way that allows continuous accumulation of values. That is, the statistics of a simulation always require the same amount of memory, regardless of the number of time steps.

6.1. Data preparation

During simulation, the receptor protein will typically exhibit translational and rotational motion. Prior to the aggregation, we therefore have to align each time step—or *snapshot*—based on the position and orientation of the protein with respect to a reference snapshot. We apply the commonly used Root Mean Square Deviation (RMSD)-minimization-based alignment [Kab76]. The alignment is executed using the simulation framework GROMACS [AMS*15], which was also used by our project partners to conduct the simulations. It tries to minimise the positional deviation of each atom to a reference position, that is, in our case, the average position of the atom over time. This is done by globally generating a rotation and a translation matrix that is applied to each atom position of a snapshot. For docking data, each found conformation of the ligand is taken as one snapshot.

6.2. Data derivation

After data preparation, all snapshots are loaded incrementally and, for each of them, several values are derived.

Distances. The most important interactions typically happen at protein atoms that are in contact with ligands for a longer time, as they might block or hinder the ligand's path to the active site. Thus, we calculate the distances between protein and ligand atoms which we can use to derive the number of close ligand atoms for each protein atom. Two atoms are close to each other if the distance between the spheres defined by their respective van-der-Waals radii is below a certain threshold value r_c . We use a default value of 3.5 Å for r_c , which was recommended by our project partners. The value delivers a good compromise, as it roughly marks the distance at which attraction between atoms induced by van-der-Waals forces becomes relevant. As the definition of closeness can depend on the use case, the threshold value can be adjusted by the user. For example, if the user wants to detect only the stronger ionic or covalent bonds, a value of 1.0 Å or even lower would be feasible. In contrast, a higher value than the default would also include atoms that may be close but do not exert attracting forces.

Bonds. Atoms of the ligand and the protein that are close to each other may form hydrogen bonds. These bonds could be either responsible for holding back a ligand or for *pulling* it forward. Hydrogen bonds can be estimated based on the element, the distance, and the bonding angle between possible bonding atoms [Jef97]. However, nonpolar ligands do not tend to form hydrogen bonds. Thus, we also incorporated carbon-carbon interactions, which are possible for

most organic molecules. This also satisfies requirement R4. These interactions are not as strong as hydrogen bonds but can still influence the movement of the ligand relative to the protein. If the distance between two carbon atoms is below the threshold $r_b \approx 3.2$ Å, a carbon-carbon interaction is assumed [ABE*09].

Landing Spots. A current hypothesis by domain scientists is that a ligand could either move directly from the surrounding medium in the active site or it could *land* on the protein surface and then crawl towards the active site. So-called *long-range-effects* of the protein surface have already been observed [LA95]. The source of those effects is unclear, may it be a protein side chain that blocks the movement of the ligand or a hydrogen bond holding it back. To provide more detailed insights into the behaviour of the ligand and to satisfy requirement R5, we extract the number of ligand landing spots on the surface of the protein. We estimate the occurrence of a landing as follows: for each ligand molecule, a counter is incremented for each snapshot if the ligand is in contact with the protein surface. If there is no contact, it is reset to zero. If one of the counters reaches the landing threshold l_t , a landing spot is registered for the currently viewed snapshot. The reason for this delayed landing spot detection is the typical behaviour of ligands, which might often only be near the protein briefly and quickly drift away again. Thus, only ligands that stay in contact for a certain time count as *landed*. As a good value for l_t depends on the time difference between the snapshots, it is a user-defined variable. In our case, a default value of 10 ps delivered the best results.

Other Values. For each protein atom, the number of different ligand molecules that contact it is calculated (n_{mol}). In case of time-dependent simulation data, the Root Mean Square Fluctuation (RMSF) is also calculated. The RMSF describes the internal per-atom movement of the protein during a simulation and is closely related to the aforementioned RMSD. It is computed by aggregating the difference between the position p_i of protein atoms i for each time step t and the corresponding atom position in a reference configuration at time step t_{ref} [VHG*18]:

$$v_{rmsf}(i) = \sqrt{\frac{1}{T} \sum_{t=1}^T (p_i(t) - p_i(t_{ref}))^2} \quad (1)$$

Again, analogous to the RMSD minimization, instead of taking a reference configuration at a certain time step t_{ref} , we use the average position of each atom as a reference. Calculating the RMSF is specifically requested by our domain experts as it provides insight into the movement of the protein that gets lost due to the aggregation.

6.3. Data aggregation

All of the derived per-snapshot data is continuously aggregated to avoid the need for later re-calculations. Please be referred to Figure 3 for further details. This leads to the values listed in Table 1. The first four values listed ($n_{contact}$, n_{ctime} , n_{bond} , and n_{btime}), as well as the landing spot count n_{spots} can be directly derived by summing up the calculated contacts and bonds. For the number of different contacted molecules n_{mol} , the ids of all contacted molecules have to be stored and summed up after all snapshots have been visited. Analogously, the continuous aggregation only calculates a part of

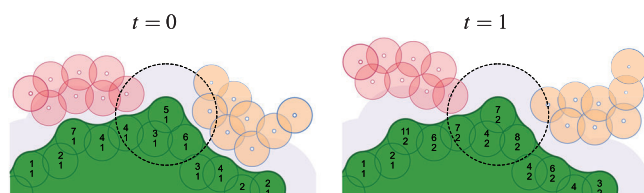


Figure 3: Aggregation of contacted atoms and time steps. In time step $t = 0$ (left), two ligands (red and orange) are in the vicinity of the protein (green). For each of the protein atoms, a radius search is performed (light blue background). For the topmost protein atom, five different ligand atoms are in the vicinity (dashed black circle). Thus, the time step counter is set to 1 (lower value) and the atom counter to 5 (upper value). Subsequently, the next simulation step $t = 1$ is loaded (right) and the same calculation is performed, adding the new values to the previously calculated ones. Note that the atoms of the protein could also be moving (not depicted).

the RMSF, namely the sum shown in Equation 1. The division by T and the taking of the root are performed after all snapshot information is known.

In addition to all of the aforementioned values, we determine the first and last time steps where contacts or bonds happen. Please note that all aggregated values divide into two categories: while members of the first category can be calculated for all considered data set types, members of the second one are only relevant for time-dependent simulation data. The members of the second category are the landing spot count, the RMSF, and the first/last time step data.

7. Visualisation of Protein-Ligand Interaction

In this chapter, we cover the design process of the developed visualisation application. It was developed in close collaboration with our domain project partners in an iterative process. The aggregation data files serve as input to the CLISD diagram (see Section 7.1), the zoomed-in view, as well as the molecular surface visualisation (see Section 7.3). The surface mesh and the protein data are only needed for the surface visualisation. As those files are small compared to the size of the actual data set and are fast to load, this enables us to satisfy requirement R7.

7.1. Compressed Ligand Interaction Sequence Diagram

The main goal of the 2D view is to give an *overview* of the protein-ligand interactions in line with requirement R1. However, this is challenging since we have many variables to represent: time steps in contact, time steps bonded, contacted atoms, bonded atoms, and RMSF (cf. Table 1). Besides, we also want to depict individual residues and provide information on bonds that occurred during the simulation. This amounts to a total of seven variables to depict in the Compressed Ligand Interaction Sequence Diagram (CLISD).

We considered different approaches, and our initial take was using a circular representation, similar to the method of Vázquez et al. [VHG*18]. However, the high number of variables rapidly saturates the central part of the plot. As a result, we decided to use a

Table 1: Overview of all values accumulated during data processing.

| Value description | Parameter name |
|-----------------------------------------|----------------|
| Number of contacted atoms | $n_{contact}$ |
| Number of time steps in contact | n_{ctime} |
| Number of bonded atoms | n_{bond} |
| Number of time steps bonded | n_{btime} |
| Number of different contacted molecules | n_{mol} |
| Landing spot count | n_{spots} |
| Root Mean Square Fluctuation (RMSF) | v_{rmsf} |
| First contacted time step | t_{fc} |
| Last contacted time step | t_{lc} |
| First bonded time step | t_{fb} |
| Last bonded time step | t_{lb} |

The upper five values can be calculated for all targeted data sets. The lower ones only apply to time-dependent simulation data.

rectangular diagram and apply some strategies to save space. First, we noted that several variables have only positive values since these are counts. As a result, we can use both the positive and negative y-axis to encode the number of time steps in contact and contacted atoms, respectively, as bars. Second, we place visual representations of the residues on the horizontal axis, similar to a classical sequence diagram. The residues are shown as small circles, colour-coded by amino acid type. We use colours that communicate their polarity, with red tones for negatively charged amino acids and blue tones for positively charged ones. Although the circles can overlap the bars, it is not a problem, since we are mostly interested in regions with high values for $n_{contact}$ and n_{ctime} , where this overlap will not matter. In a second step, we use superposition to add more variables. Thinner bars with an elliptic cap encode the bonded time steps and bonded atoms in the positive and negative areas, respectively. Finally, we draw the RMSF as a line chart in the positive area, leading to six additional variables encoded besides the sequence itself. The complete map is shown in Figure 4.

By using larger thickness and less saturated colours for the bars in the background, and more saturated colours and opacity for the elements in the foreground, we are able to generate an information-rich view, as shown in Figure 4. Note that all values have vastly different ranges, therefore, we do not show a y-axis with unit markers. Due to these range differences, bringing them on the same scale would make many bars unreadable small. Even under the consideration to scale some bars logarithmically where needed, it would require at least two different logarithmic scales. Multiple logarithmic scales, however, could be even more confusing to the viewer than scaling all values separately. Therefore, the CLISD only gives a qualitative overview of the value distribution. However, if the user hovers over a residue, a tooltip-like text box with the quantitative values appears, enabling a detailed analysis. If the user selects a residue, the values are shown in the top centre view (cf. Figure 5).

Since not all residues are equally important for researchers (e.g. residues that exhibit a high number of ligand contacts have a more prominent effect on the interaction), we filter based on this data. However, instead of completely erasing non-contacting residues, they are assigned a smaller amount of space in horizontal direction. This way, we avoid producing false impressions of the positions

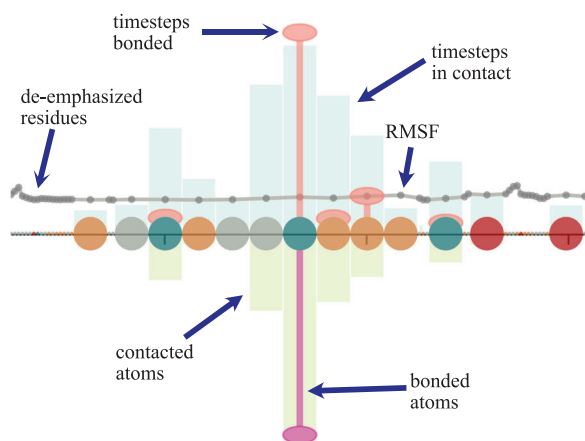


Figure 4: The variables represented in the CLISD. Bars in the upper half represent counted time steps, bars in the lower half counted atoms. When bonds are detected, a narrower bar is used as representation. Where applicable, the RMSF is displayed as gray line in the upper half. Areas where only few contacts are detected, are de-emphasised by shrinking them in x-direction.

of the residues in the sequence. Since the majority of residues will never be in contact with ligands during the simulation, this makes our diagram also more scalable than a simple equally-spaced distribution of all the elements. This is shown on the left-hand side of Figure 4.

Although we designed the CLISD in a way that it should be easily readable, there could be situations in which it could appear too cluttered. Therefore, we decided to stack multiple instances of a CLISD, enabling the user to switch off certain variables arbitrarily. As we worked with a default 16:9 aspect ratio, we typically used two diagrams. For other ratios, more diagrams can be ideal.

7.2. Sequence detail view


In addition to the main view showing the CLISD, we added a detail view. An example of this detail view is shown in the top right of Figure 5. The detail view does not de-emphasise low-contacted residues and shows only a window of a few residues. Please note the small width of the scroll bar in Figure 5, indicating that a complete display of this diagram would easily exceed the available screen space. This zoomed in view is added since the de-emphasised residues might directly influence the emphasised ones, for example by performing large movements indicated by high RMSF values. The detail view is linked to the main view: if the user selects a residue, the zoomed-in detail view is able to centre on the same residue, allowing for an in-depth analysis of this section of the protein even if the residues are de-emphasised in the main view.

7.3. Molecular surface visualisation

The SES of the protein received from the preprocessing step is stored as a triangle mesh. Each mesh vertex has multiple properties (e.g. colours, normals for lighting) required for rendering.



Figure 5: Screenshot of the complete application. In the upper left, the surface visualisation can be seen. In top central, general information about the visualised protein is displayed, alongside with the selection for the data set as well as the currently displayed variable. Below that, the exact calculated values for the currently selected amino acid can be inspected. The lower half shows the CLISD with a mouseover text. In the upper right, the zoomed in view of the CLISD is displayed. Besides the 3D view and the zoomed in view a legend is displayed.

Eight different colouring modes are available that either highlight the individual elements (amino acids) of the protein or to colour-map the aggregated values of the protein-ligand interaction onto the surface. The colouring modes show the values $n_{contact}$, n_{time} , n_{bond} , n_{btime} , n_{mol} , n_{spots} , and n_{rmsf} . Additionally, an eighth colouring mode depicts the hydrophobicity of the underlying amino acids. The first seven modes have their colour scale in common. This reaches from violet for zero over green for the mid value to yellow for the maximum value:  (see Figure 1 or Figure 5). This sequential colour map is also known as the Viridis colour map, which is one of the default colour maps of the Python plotting library Matplotlib [Hun07]. Liu and Heer [LH18] showed that this colour map is superior to several other sequential colour maps in terms of error rates and response times. The hydrophobicity colouring mode depicts hydrophilic amino acids in blue and hydrophobic ones in yellow. In contrast to the sequence diagram, the surface visualisation uses the aggregated per-atom values instead of the values per amino acid. These values are more detailed and allow for a better perimeter of the relevant surface areas.

7.4. Linked interaction

As mentioned above, all views of our web-based visualisation application are linked. If a user selects an atom in the 3D view, the whole residue will be selected on the mesh and will also be marked in the CLISD as well as in the zoomed-in view. The opposite direction—selecting a residue in any of the diagrams and highlighting it in the mesh—is also possible. Selections on the mesh are shown via a semi-transparent orange overlay. In the CLISD, a vertical line is put above the x-axis at the location of the corresponding residue. Hovering over the diagram or the surface has a similar effect, but we use another colour to show hovering interactions. With multiple CLISD diagrams, these marks are also propagated to the additional instances. Additionally, as requested by our domain ex-

perts, the user is allowed to enter multiple amino acid identifiers that get marked separately. With this feature, e.g. marking the active site permanently for further orientation becomes possible. The necessary identifiers can be found out by using the mouse-over texts, but they are typically known by the domain experts beforehand.

8. Implementation Details

As mentioned above, our proposed framework consists of two separate applications for data processing and visualisation. In this section, we briefly describe how these applications were implemented and which existing libraries and frameworks were used for the development.

8.1. Data processing application

The application for value aggregation was written in C++ using the visualisation framework MegaMol [GKM*15]. For the fixed-radius neighbour search, we use the KD-tree implementation of *nanoflann* [BR14]. The KD-tree construction has to be performed for each snapshot. Since there are usually much fewer ligand atoms than protein atoms, we store the ligand atoms in the tree. This is also beneficial for the parallelization of the aggregation process, as it enables a lock-free parallel computation for all protein atoms. A parallelization over all frames would also be possible, but as we restricted ourselves only to commodity machines, our approach is mostly IO-capped (cf. Section 9.3). This means that such a parallelization strategy could even harm the data throughput as the data is not read continuously.

The input data is given in the Protein Data Bank (PDB) file format [BWF*00]. As this format is text-based and not designed to represent trajectory data, most simulation programs implement their own file format to store the individual time steps of the simulation. In case of the MD simulation framework *GROMACS* [VDSLH*05], this is the binary XTC trajectory file format.

The aforementioned calculations are performed for each atom of the protein and each amino acid. The amino acid values cannot be simply obtained by aggregating the already aggregated per-atom values, as this would lead to erroneous values due to multiple counting of contacts. All results of the aggregation calculations are written to a structured *json* file, which is later used for visualisation. The *json* file contains an entry for each amino acid, listing the identifier, the amino acid type, and all aggregated values.

The 3D visualisation requires a protein configuration. While this is simple for docking data (as there is only one configuration of the protein), a simulation of n time steps also offers n plausible choices for a displayable configuration. To retrieve the most representative configuration, we first compute the average structure of all atom positions and then find the time step that has the smallest RMSD [Kab76] to that average. Additionally, our visualisation requires a SES mesh, which is calculated via the *MSMS* tool [SOS96] and stored as Polygon File Format (PLY) file, allowing an arbitrary number of vertex attributes. Thus, we store all aggregated values alongside with atom identifiers for each vertex. As the visualisation additionally requires a PDB file, one is generated by copying

the input PDB file and interchanging the atom coordinates with the coordinates of the representative time step.

We intentionally chose popular, standardised formats (PDB, JSON, PLY) as it makes the calculated data easy to read, even without our visualisation application. This enables domain scientists to use the data more freely and increases reproducibility.

8.2. Web-based visualisation application

As mentioned above, our visualisation application is purely web-based. A screenshot of the application is depicted in Figure 5. It uses a server-client architecture, is written in typescript, and uses WebGL for 3D visualisation, whereas the CLISD is written in JavaScript using D3 [BOH11]. All views are linked with each other, allowing the user to select an amino acid in any of them. This selection is propagated to all other views.

To load different data sets, a drop-down menu above the CLISD was chosen. If a new set is selected the diagram will update automatically and the corresponding PLY and PDB files are loaded. The PDB file is needed to assign the atom IDs from the mesh to the amino acids. In another drop-down menu, the different aggregated variables for the surface colouring can be selected. By request of our domain scientists, a colour legend, as well as legend for the CLISD, is rendered to provide more orientation.

8.2.1. Sequence diagram layout

To improve the usage of space, we de-emphasise residues with a low number of contacts by assigning them less space. Instead of fixing a constant size for those, we calculate it dynamically for each model. This way, we ensure that most of the horizontal sequence space is devoted to important residues. Thus, we fix a parameter $W_{ER} \in [0, 1]$ that determines the total width for all emphasised residues, and the remaining space is dedicated to the remaining residues. The first task is to count the number of residues to emphasise (n_e). Given a contact threshold n_c defined by the user, all amino acids with a $n_{time} < n_c$ are de-emphasised. Then, given the space we want to devote to the important residues W_{ER} , the size of each emphasised residue w_{er} will be:

$$w_{er} = width * W_{ER} / n_e \quad (2)$$

After some experimentation, we empirically found that a value of $W_{ER} = 0.7$ of the total available space was suitable in most cases. This produces the result shown in Figure 4. Larger values for W_{ER} would not visually filter the residues effectively, while much lower values would make them imperceptible.

8.2.2. Molecular surface mesh rendering

To display the generated acSES mesh, we build upon the Mol* library (<http://molstar.org>), a collaborative project by the PDB in Europe and the RCSB PDB. It provides a technology stack for data delivery and analysis tools for macromolecules. Apart from file loading for many biomolecular file formats, it offers various common molecular visualisations such as ball-and-stick. To properly incorporate our precalculated data, we extended the library to support

reading PLY files (see Section 7.3), including all embedded vertex properties for colouring and grouping. This results in the surface visualisation depicted in the top left of Figure 5. Note that the probe radius for the shown SES is already determined by the user during the precalculation step. It should typically correspond to the size of the ligands. The rendered mesh supports picking of vertex groups, which is used to identify the corresponding residues upon hovering over them or selecting them with the mouse. Due to the use of Mol* it is possible for the user to interactively change the lighting conditions and add effects like ambient occlusion or the marking of outlines.

9. Results and Discussion

To evaluate our aggregation and visualisation system, we present two different application cases. We received the presented data sets from different collaboration partners working in the fields of biochemistry and bioinformatics. The first data set (Section 9.1) is a simulation ensemble consisting of 10 independent runs. We evaluated both the whole ensemble as well as all runs separately. The second data set (Section 9.2) is a much smaller ligand docking set. While the creators of the first data set just provided the data, the creators of the first one actively collaborated in the design process of the visualisation. In Section 9.3, we describe the performance aspects of our approach.

9.1. Application case I: MD-simulation

Goal. One of the goals of the MD-simulation ensemble was to study the pathways of substrate molecules under realistic conditions. This especially includes a study of the pathways of the substrate molecules on the surface of the receptor protein, as already mentioned in Section 6. So researchers want to find out whether molecules interacting with the active site have landed there from the solvent, or after previous contact(s) on the surface, and if so, how the interactions happen. With current techniques, such as the filtering approach proposed by Vad et al. [VBJ*17], some of this information can be obtained, but the spatial inspection would have been particularly difficult due to the size of the data set. Especially for the complete ensemble, millions of line segments would have to be rendered, even after filtering. This quickly becomes unfeasible performance-wise, as well as in terms of visual clutter. Therefore, we decided against a direct visualisation approach to satisfy requirement R3. And other advanced features we provide, like the identification of landing spots would be even more difficult with other methods.

Input Data. The simulation data set contains an ensemble simulation of a mutated variant of the enzyme *Candida antarctica* lipase B (CALB) [UHPJ94], comprising 10 independent runs of 160 ns each. With a time step size of 1 ps, the total number of simulation time steps over all runs is 1.6 million. The simulated region is a cubic space (side length 32 nm) containing one CALB protein in water, alongside with 20 realistically distributed 4-paranitrophenol substrate molecules. The simulation was performed using *GROMACS* [AMS*15], resulting in a final binary data set size of ~2 TB, consisting mostly of water. Water molecules in the input data were

filtered out during the data preprocessing step as it was only required to create a realistic environment.

Analysis and Results. On a commodity machine (cf. requirement R6), the precomputation/aggregation step takes roughly 4.2 hours for all ten runs (see Section 9.3 for more details). Our project partners appreciated this, as the actual simulation, in comparison, took several days. As their goal was to study general ligand pathways, the precalculated landing spot values were of immense importance to them as they provide hints where the ligand molecules come in contact first. Setting the landing threshold value l_t to 10 ps (see Section 6) was a request by them. This value provides a good compromise between a stable contact and a good spatial correlation between measured and actual landing spots. When checking their hypothesis that ligand molecules do not directly access the active site but *crawl* along the surface, it quickly became clear that this is mostly the case. One example of this behaviour is shown in Figure 6. It depicts the molecular surface of the fourth ensemble run. Viewing the number of contacted atoms $n_{contact}$, a lot of movement seems to happen around the cavity containing the active site, indicating that it was indeed accessed. The hypothesis that the molecules moving there did not directly *fly* in from the surrounding medium can be tested by switching to the n_{spats} colouring scheme (requirement R5), which affirms the hypothesis, as no landing spots seem to be detected at the active site. The exact values can be checked in the linked CLISD. By inspecting the surface, one can clearly identify two ligand pathways to or from the active site which also satisfies requirement R3 in an implicit manner. Such phenomena can be observed in most of the simulation runs, although the active site is not reached in all of them. The $n_{contact}$ and n_{ctime} view of the complete simulation ensemble even shows that the path marked with the number 3 in Figure 6(a) is a common path across all runs. By checking the proportions between the $n_{contact}$ and n_{ctime} value and comparing them with the values of other amino acids, one can derive the directness of the contacts. Where $n_{contact}$ is approximately the same as n_{ctime} , only slight contact is made, meaning that only few atoms of the ligand are close to the surface. If $n_{contact}$ is far larger than n_{ctime} , more direct contact can be assumed, as more atoms of the ligand are closer to the viewed protein atom. Further inspection of other colouring modes leads to the insight that these movements were performed by multiple ligand molecules.

In addition to the pathways, another observation can be made. Our project partners specifically requested the calculation of carbon-carbon interactions, as the inspected ligand is nonpolar (cf. R4). When checking the bond variables n_{bond} and n_{ptime} , it becomes obvious that most of the carbon-carbon interactions happen while the ligand resides in a *valley* of the surface. In conjunction with n_{ctime} , the conclusion is obvious that some valleys seem to hold the ligand molecules in by forming bonds. With that knowledge, it is a possibility to engineer another mutant of the protein that does not contain such valleys.

Discussion. When confronted with the visualisation framework, our project partners who conducted the simulation were especially fond of the interaction possibilities. However, to understand the visualisation fully, it was necessary to explain that the x-axis of the sequence diagram indeed depicts the amino acid sequence. After their feedback, we added legends to our visualisation and included the possibility to permanently mark certain amino acids. Additionally, we

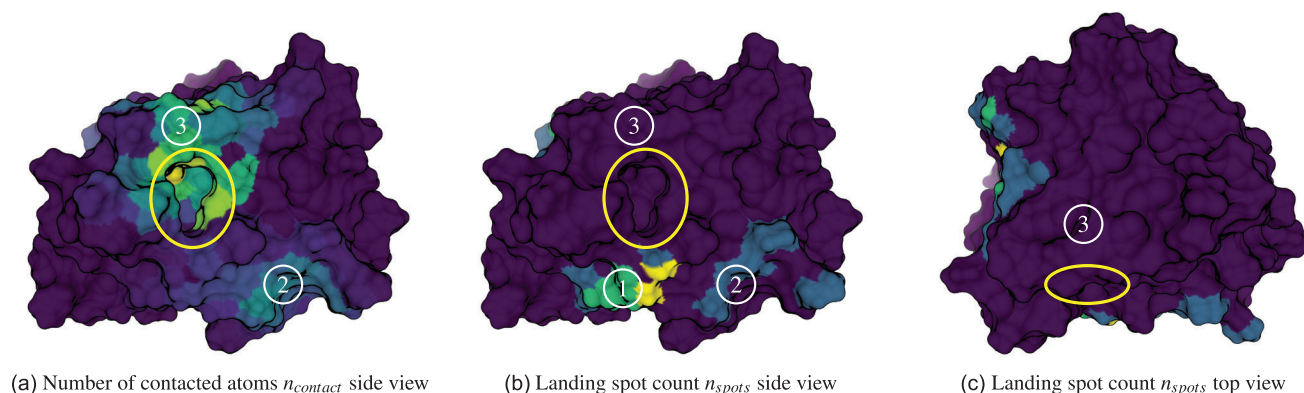


Figure 6: Surface visualisation of the fourth run of the ensemble data set. The cavity containing the active site is marked with a yellow circle. Images (b) and (c) show the same surface but from different angles. All of the detected landing spots (marks 1 and 2) are not inside or at the border of the active site cavity. The $n_{contact}$ mesh (a) reveals that there was indeed ligand movement towards and from the active site (mark 3). Additionally, it can be seen that ligands tend to stay rather long in the valley at mark 2.

tweaked the wording shown in the GUI to better meet their expectations. They also noted one minor drawback: As not all residues are part of the surface all the time, no area may be marked in the surface visualisation when clicking on a residue in the sequence view. This behaviour is mitigated by the de-emphasis of residues without contact. Additionally, our collaboration partners liked the fact that they did not need to use additional software and could share the visualisation with colleagues by simply sending them a link. Unexpected by us, they also appreciated that the rotation of the surface visualisation stayed the same when switching colouring modes, helping them to maintain orientation.

9.2. Application case II: Molecular docking experiment

Goal. Besides simulation data, our tool can also be used to analyse docking results. Molecular docking experiments try to evaluate possible and energetically favourable ligand orientations and positions on the protein's surface. It is a frequently used method in drug design, where the applicability of a newly developed drug (the ligand) is tested for a target protein (the receptor). The aim is to predict locations on the surface where the ligand will bind most probably. However, the opposite way is also possible. For computationally modeled proteins, the active sites for given ligands are not always known beforehand or are not experimentally verified. Docking can help to discover all available active sites of the modeled protein. Other existing systems for the investigation of molecular docking, like the one of Seeliger and de Groot [SdG10], predominantly show only single ligands at a time. This does not allow for the detection of general tendencies and may even be misleading, as the most energetically favourable ligand position might be an outlier. To our knowledge, our tool is currently the only one to give a protein-centric overview of all ligand conformations.

Input Data. The data set presented in this section falls in the latter category. Multiple ligands were tested against a given receptor protein. For each ligand, several runs were performed, without specifying the target position on the surface of the protein. Each of the runs comprises one hundred possible ligand conformations. In our

preprocessing step, each run was handled separately. As mentioned in Section 6, not all of the calculated variables are applicable when using docking data. While the upper five values of Table 1 work as intended, none of the lower six is applicable. Since docking results have no temporal component, calculation of time step-related variables is impossible. Due to the missing movement of the molecules, neither landing spots nor the RMSF can be calculated.

Analysis and Results. After the preprocessing, domain experts used our tool to explore the results. Figure 7 shows a cutout of the CLISD, as well as the most relevant part of the protein surface of one of the runs. In this case, hydrogen bonds were chosen as bond types. Combining all information, three specific areas of interest can be identified. The first is the area with the highest ligand presence probability, namely the residue with the highest n_{ctime} value. Most of the tested ligand molecules (82 out of 100) were docked in this location, so some of the surface properties of the protein are highly likely to be favourable for ligand docking. One of these reasons might be a hydrogen bond formed at one certain location of the surface, marked with a red border. The presence of this bond can be investigated in the sequence view or the surface view by switching to one of the two bond colouring modes. Our linked views allow for a direct switch between the two representations. The third area of interest is on the upper left of the surface view, marked in cyan. Only few ligand molecules found there an energetically favourable position to bind, leading to a slightly lighter tone of purple as surface colour. When investigating this further, the lack of hydrogen bonds in this location might be one of the reasons.

An additional feature of our visualisation is the easy identification of outliers in the surface view. The visualised protein contains multiple areas where ligands find a position to dock. In fact, there is one on the other side of the protein, represented by the violet marked bars on the far left of the CLISD. Depending on the application case, such areas can be ignored or can be seen as a possible target for mutations if one wants to optimise the reaction rates.

Discussion. For this concrete problem, ligand docking analysis, our tool allows for an easy and direct visual identification of relevant

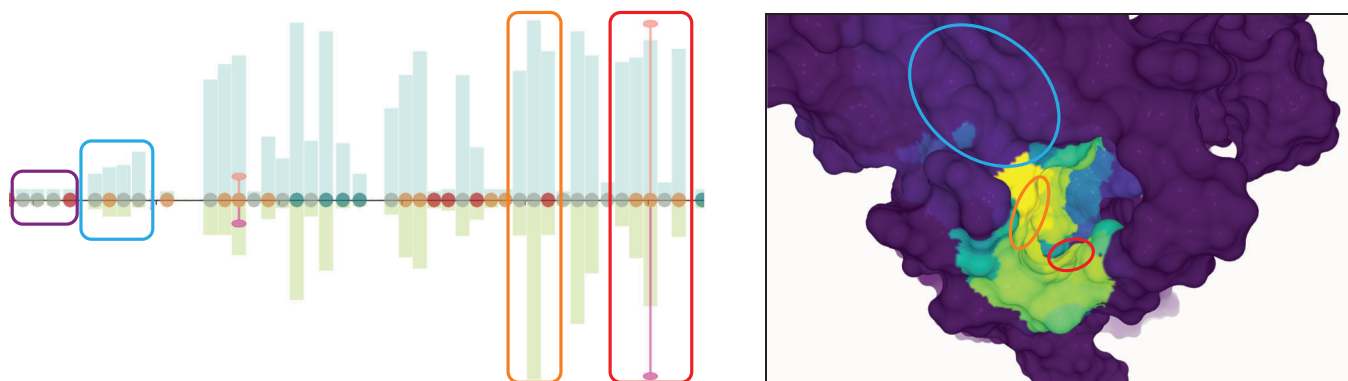


Figure 7: CLISD and surface visualisation showing the number of contacted time steps n_{ctime} of one single run of the molecular docking data set. As docking only occurs in a specific area, both views have been cropped to these parts. In the surface view, a predominant docking area, marked as green and yellow surface colour, can be identified. The highest contact values ($n_{contact} = 1168$, $n_{ctime} = 82$, orange marks) occur there. Only few actual hydrogen bonds are formed, but the most often occurring one ($n_{bond} = 20$, red marks) is inside the crowded area. This hydrogen bond may be one of the causes for the ligand's tendency to bind at this location. Besides the crowded area, an area with fewer contacts without hydrogen bonding can be identified (cyan marks).

Table 2: Data set sizes before and after aggregation.

| Data set | Original | Json | PDB | Mesh |
|---------------------|----------|--------|--------|-------|
| Simulation (1 run) | 200 GB | 68 KB | 357 KB | 16 MB |
| Simulation ensemble | 2 TB | 68 KB | 357 KB | 16 MB |
| Docking (1 run) | 65 MB | 112 KB | 623 KB | 35 MB |

Cursive values mark binary data sets, the other ASCII-encoded ones. As the simulation run is part of the simulation ensemble, the size of the resulting data stays the same, although it contains different values.

areas that may be prone to ligand binding. These specific input data sets are also a good example for the *other uses* shown in Figure 2. In addition to this, using the standardised output data, our collaboration partners were able to work with the aggregated results even before the visualisation system was even usable. They used MegaMol (other frameworks for protein surface visualisation were also possible) to generate surface renderings for an upfront analysis of the ligand density distribution.

9.3. Performance and scalability

For a data aggregation approach, not only the running times but also the data set sizes are of importance. The smaller the final size is, the easier it is to transfer and visualise. This, of course, also means a higher data loss. The omitted data is in our case mainly the absolute atom position of each atom and the accountability of specific ligands to the observed effects. Resulting data set sizes are shown in Table 2. It is observable that the resulting sizes heavily depend on the size of a single snapshot. Although the ligand docking set was much smaller in size than the other two, the resulting data is roughly twice as large. The simulated protein consisted out of ~ 300 , and the ligand docking receptor protein out of ~ 500 amino acids. As the single simulation run and the whole ensemble work on the same data, the final data size is the same for both. This leads to compression by five orders of magnitude in the case of the ensemble. With

our approach being agnostic about the length of the simulation, it is possible to process Petabytes of input data without increasing the memory footprint. Please note that the mesh data could be further compressed, as we chose the more easily writable ASCII-encoded version of PLY instead of the binary variant.

As the aggregation times for the ligand docking data set were dominated by startup overhead, we evaluated them for the first run of the simulation data. The initial reading of the data to retrieve the average atom positions to later on calculate the RMSF value took 699 s, the second run through the data for actual value aggregation took 917 s. Timings do not include the writing time of the output data. This results in a data throughput of about 150 MB/s, which is close to the maximum reading speed of the used hard drive. The results for the other runs are roughly the same, differing only by a few seconds. Aggregation of the whole ensemble takes straightforwardly ten times longer than the calculation of one of the ten ensemble members.

The final rendering of the aggregated values leads to no performance problems whatsoever. We evaluated our web-based tool in four different browsers (Google Chrome, Firefox, Opera, and Microsoft Edge). As it was not possible to turn off VSync for all browsers, we left it on for all measurements. It appeared that the visualisation reaches almost constant 60 fps, even with a moving surface visualisation. Only small drops to roughly 30 fps were noticeable as the user changes the input data set or performs a brushing operation. Although our application also runs smoothly in modern smartphone browsers, we recommend using at least a tablet to view the results, due to the larger available screen size.

As shown, with the used data sets the preprocessing scales well with the length of the data set. We could not observe effects like a changing size of a single time step, which could influence the preprocessing time in a non-linear manner, as the used data was not diverse enough. As stated, the IO time was dominant in our case, but with increasing sizes for single time steps the radius search for ligand atoms could become dominant as it scales with $\mathcal{O}(m \log n)$

(m : no. of protein atoms, n : no. of ligand atoms) and not linearly. Although the actual visualisation has no performance issues, available screen size could be a limiting factor for the CLISD. While our approach scales better than the one of Vázquez et al. [VHG*18], for example, too small bar widths can render the visualisation unreadable. In our opinion, this limit is reached when the width for individual bars reaches values below 1 mm and the dividing room between them values below 0.5 mm. For a screen width of 50 cm this is the case when at least 233 amino acids have to be rendered emphasised (still assuming that $W_{ER} = 0.7$). This can only be reached with either very large proteins or evenly distributed ligand movement over the complete surface. The latter was the case in the combined 10 run dataset but the screen space issued to the emphasised amino acids was still more than double than the specified limit.

10. Summary and Future Work

We presented an approach for the visual analysis of very large protein-ligand interaction data sets. Our approach includes a data aggregation pipeline as well as an interactive web-based visualisation framework, which combines novel and traditional visualisations, namely an enhanced sequence diagram that supports level-of-detailing (the CLISD) and a commonly used molecular surface (the SES) of the targeted protein. Aggregating the values of interest, like the number of contacted ligand molecules or the number of found ligand landing spots, results in a data representation that is completely independent of the number of snapshots of the underlying data set. Our web-based framework is then able to visualise the overview of the simulation by adding as many variables as demanded. As mentioned above, the users can decide either to show all the aggregated data or filter out some of them. Since our application is lightweight and uses only standard web technologies like HTML5 and JavaScript. Therefore, and since the aggregated data has a small memory footprint, it can even be used on mobile devices like smartphones or tablets. Domain scientists can use it to inspect the surface of the receptor protein in the surface visualisation view. Via brushing and linking, it is possible to identify amino acids that interact with ligands in the novel abstract sequence diagram. Our enhanced sequence diagram de-emphasises low-value areas by decreasing their size and, therefore, directs the attention of the user towards the more important values. A zoomed-in detail view shows the full information for each amino acid if required. We demonstrated the utility of the aggregation and visualisation using an individual MD simulation and a simulation ensemble, as well as ligand docking experiments.

In the future, we plan to incorporate aggregated ligand paths to foster the understanding of the movement. This approach has already been used for water molecules in simulations, for example by Bidmon et al. [BGB*08] or by Vad et al. [VBJ*17], and would complement our surface-based visualisation. Another possible extension would be to incorporate additional interaction forces apart from the three currently used ones (range-based, hydrogen bonds, and carbon-carbon interactions). van der Waals forces, for example, become relevant when the distance between atoms becomes smaller. However, vdW forces are not as easily parameterizable as other force types, which makes their calculation more difficult. Additionally, a further incorporation of physico-chemical properties of the surface into the precomputation could deliver more exact results.

Finally, an extension to protein-protein interactions may also be useful to domain scientists.

Acknowledgements

The authors wish to thank Ragothaman Yennamali for the protein docking data set. This work has been partially funded by *German Research Foundation* (DFG) as project PROLINT (project number 391088465), and project TIN2017-88515-C2-1-R (GEN3DLIVE), from the *Spanish Ministerio de Economía y Competitividad*, by 839 FEDER (EU) funds. M.K. was funded by Carl-Zeiss-Stiftung. The RCSB PDB was jointly funded by the NSF, the NIH and the US DoE [NSF DBI-1338415; PI: SK Burley].

Open access funding enabled and organized by Projekt DEAL.

References

- [AAM*17] ALHARBI N., ALHARBI M., MARTINEZ X., KRONE M., ROSE A., BAADEN M., LARAMEE R. S., CHAVENT M.: Molecular visualization of computational biology data: A survey of surveys. In *EuroVis - Short Papers* (2017), vol. 1, pp. 133–137. <https://doi.org/10.2312/eurovisshort.20171146>.
- [AAM*17] ALKORTA I., BLANCO F., ELGUERO J., DOBADO J. A., FERRER S. M., VIDAL I. (2009) Carbon...Carbon Weak Interactions. *The Journal of Physical Chemistry A*, 113 (29), 8387–8393. <https://doi.org/10.1021/jp903016e>.
- [AKCL19] ALHARBI N., KRONE M., CHAVENT M., LARAMEE R. S.: Hybrid Visualization of Protein-Lipid and Protein-Protein Interaction. In *Proc. EG VCBM* (2019), 213–223. <https://doi.org/10.2312/vcbm.20191247>.
- [ALC16] ALHARBI N., LARAMEE R. S., CHAVENT M.: Mol-PathFinder: interactive multi-dimensional path filtering of molecular dynamics simulation data. *Conference on Computer Graphics & Visual Computing* (2016), 9–16. <https://doi.org/10.2312/cgvc.20161289>.
- [AMS*15] ABRAHAM M. J., MURTOLA T., SCHULZ R., PÁLL S., SMITH J. C., HESS B., LINDAHL E.: GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* 1 (2015), 19–25. <https://doi.org/10.1016/j.softx.2015.06.001>.
- [BGB*08] BIDMON K., GROTTTEL S., BÖS F., PLEISS J., ERTL T.: Visual abstractions of solvent pathlines near protein cavities. *Computer Graphics Forum* 27, 3 (2008), 935–942. <https://doi.org/10.1111/j.1467-8659.2008.01227.x>.
- [BJG*15] BYŠKA J., JURČIK A., GRÖLLER M. E., VIOLA I., KOZLÍKOVÁ B.: MoleCollar and tunnel heat map visualizations for conveying spatio-temporo-chemical properties across and along protein voids. *Computer Graphics Forum* 34, 3 (2015), 1–10. <https://doi.org/10.1111/cgf.12612>.
- [BLMG*16] BYŠKA J., LE MUZIC M., GRÖLLER M. E., VIOLA I., KOZLÍKOVÁ B.: AnimoAminoMiner: Exploration of protein tunnels and their properties in molecular dynamics. *IEEE Trans-*

- actions on *Visualization and Computer Graphics* 22, 1 (2016), 747–756. <https://doi.org/10.1109/TVCG.2015.2467434>.
- [BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309. <https://doi.org/10.1109/TVCG.2011.185>.
- [BR14] BLANCO J. L., RAI P. K.: nanoflann: a C++ header-only fork of FLANN, a library for nearest neighbor (NN) with kd-trees, 2014. URL: <https://github.com/jlblancoc/nanoflann>.
- [BTM*19] BYŠKA J., TRAUTNER T., MARQUES S. M., DAMBORSKÝ J., KOZLÍKOVÁ B., WALDNER M.: Analysis of Long Molecular Dynamics Simulations Using Interactive Focus+Context Visualization. *Computer Graphics Forum* 38, 3 (2019), 441–453. <https://doi.org/10.1111/cgf.13701>.
- [BWF*00] BERMAN H. M., WESTBROOK J., FENG Z., GILLILAND G., BHAT T. N., WEISSIG H., SHINDYALOV I. N., BOURNE P. E.: The Protein Data Bank. *Nucleic Acids Research* 28, 1 (2000), 235–242. URL: <http://www.pdb.org>.
- [Con83] CONNOLLY M. L.: Analytical molecular surface calculation. *J. App. Crystallogr.* 16, 5 (1983), 548–558. <https://doi.org/10.1107/S0021889883010985>.
- [DHR*19] DURAN D., HERMOSILLA P., ROPINSKI T., KOZLÍKOVÁ B., VINACUA A., VÁZQUEZ P.-P.: Visualization of large molecular trajectories. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 987–996. <https://doi.org/10.1109/TVCG.2018.2864851>.
- [Fis94] FISCHER E.: Einfluss der Configuration auf die Wirkung der Enzyme. *Ber. Dtsch. Chem. Ges.* 27, 3 (1894), 2985–2993. <https://doi.org/10.1002/cber.18940270364>.
- [FJB*17] FURMANOVÁ K., JAREŠOVÁ M., BYŠKA J., JURČÍK A., PARULEK J., HAUSER H., KOZLÍKOVÁ B.: Interactive exploration of ligand transportation through protein tunnels. *BMC Bioinformatics* 18, 2 (2017), 22. <https://doi.org/10.1186/s12859-016-1448-0>.
- [FJK*19] FURMANOVÁ K., JURČÍK A., KOZLÍKOVÁ B., HAUSER H., BYŠKA J.: Multiscale visual drilldown for the analysis of large ensembles of multi-body protein complexes. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 843–852. <https://doi.org/10.1109/TVCG.2019.2934333>.
- [GKM*15] GROTTTEL S., KRONE M., MÜLLER C., REINA G., ERTL T.: MegaMol – a prototyping framework for particle-based visualization. *IEEE Transactions on Visualization and Computer Graphics* 21, 2 (2015), 201–214. <https://doi.org/10.1109/TVCG.2014.2350479>.
- [GLB*16] GOWERS R. J., LINKE M., BARNOUD J., REDDY T. J. E., MELO M. N., SEYLER S. L., DOMAŃSKI J., DOTSON D. L., BUCHOUX S., KENNEY I. M., BECKSTEIN O.: MDAnalysis: A python package for the rapid analysis of molecular dynamics simulations. In *Proceedings of the 15th Python in Science Conference* (2016), pp. 98–105. <https://doi.org/10.25080/Majora-629e541a-00e>.
- [HDS96] HUMPHREY W., DALKE A., SCHULTEN K.: VMD: Visual Molecular Dynamics. *J. Mol. Graph.* 14, 1 (1996), 33–38. [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5).
- [HEG*17] HERMOSILLA P., ESTRADA J., GUALLAR V., ROPINSKI T., VINACUA À., VÁZQUEZ P.-P.: Physics-based visual characterization of molecular interaction forces. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 731–740. <https://doi.org/10.1109/TVCG.2016.2598825>.
- [HS12] HEER J., SHNEIDERMAN B.: Interactive dynamics for visual analysis. *ACM Queue* 10, 2 (2012), 30–55. <https://doi.org/10.1145/2133806.2133821>.
- [Hun07] HUNTER J. D.: Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. <https://doi.org/10.1109/MCSE.2007.55>.
- [IWR*18] IBRAHIM M., WICKENHÄUSER P., RAUTEK P., REINA G., HADWIGER M.: Screen-space normal distribution function caching for consistent multi-resolution rendering of large particle data. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 944–953. <https://doi.org/10.1109/TVCG.2017.2743979>.
- [Jef97] JEFFREY G. A.: *An Introduction to Hydrogen Bonding*. Topics in Physical Chemistry. Oxford University Press, 1997.
- [Kab76] KABSCH W.: A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. A* 32, 5 (1976), 922–923. <https://doi.org/10.1107/S0567739476001873>.
- [KKF*16] KOZLÍKOVÁ B., KRONE M., FALK M., LINDOW N., BAADEN M., BAUM D., VIOLA I., PARULEK J., HEGE H.-C.: Visualization of biomolecular structures: State of the art revisited. *Computer Graphics Forum* 36, 8 (2016), 178–204. <https://doi.org/10.1111/cgf.13072>.
- [KKL*16] KRONE M., KOZLÍKOVÁ B., LINDOW N., BAADEN M., BAUM D., PARULEK J., HEGE H.-C., VIOLA I.: Visual analysis of biomolecular cavities: State of the art. *Computer Graphics Forum* 35, 3 (2016), 527–551. <https://doi.org/10.1111/cgf.12928>.
- [KWN*14] KNOLL A., WALD I., NAVRATIL P., BOWEN A., REDA K., PAPKA M. E., GAITHER K.: RBF volume ray casting on multicore and manycore CPUs. *Computer Graphics Forum* 33, 3 (2014), 71–80. <https://doi.org/10.1111/cgf.12363>.
- [LA95] LICATA V. J., ACKERS G. K.: Long-Range, Small Magnitude Nonadditivity of Mutational Effects in Proteins. *Biochemistry* 34, 10 (1995), 3133–3139. <https://doi.org/10.1021/bi00010a001>.
- [LH18] LIU Y., HEER J.: Somewhere over the rainbow: An empirical assessment of quantitative colormaps. In *Proc. of the CHI Conference on Human Factors in Computing Systems* (2018), pp. 598:1–598:12. <https://doi.org/10.1145/3173574.3174172>.

- [LMA*18] LICHTENBERG N., MENGES R., AGEEV V., GEORGE A., HEIMER P., IMHOF D., LAWONN K.: Analyzing residue surface proximity to interpret molecular dynamics. *Computer Graphics Forum* 37, 3 (2018), 379–390. <https://doi.org/10.1111/cgf.13427>.
- [LMPV15] LE MUZIC M., AUTIN L., PARULEK J., VIOLA I.: celVIEW: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In Proc. EG VCBM (2015), The Eurographics Association, pp. 61–70. <https://doi.org/10.2312/vcbm.20151209>.
- [LR71] LEE B., RICHARDS F. M.: The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol.* 55, 3 (1971), 379–380. [https://doi.org/10.1016/0022-2836\(71\)90324-X](https://doi.org/10.1016/0022-2836(71)90324-X).
- [MDLI*18] MIAO H., DE LLANO E., ISENBERG T., GRÖLLER M. E., BARISIC I., VIOLA I.: DimSUM: Dimension and Scale Unifying Maps for visual abstraction of DNA origami structures. *Computer Graphics Forum* 37, 3 (2018), 403–413. <https://doi.org/10.1111/cgf.13429>.
- [MDWB11] MICHAUD-AGRAWAL N., DENNING E. J., WOOLF T. B., BECKSTEIN O.: MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.* 32, 10 (2011), 2319–2327. <https://doi.org/10.1002/jcc.21787>.
- [PJR*14] PARULEK J., JÖNSSON D., ROPINSKI T., BRUCKNER S., YNNERMAN A., VIOLA I.: Continuous levels-of-detail and visual abstraction for seamless molecular visualization. *Computer Graphics Forum* 33, 6 (2014), 276–287. <https://doi.org/10.1111/cgf.12349>.
- [PRV13] PARULEK J., ROPINSKI T., VIOLA I.: Seamless visual abstraction of molecular surfaces. In Proceedings of the 29th Spring Conference on Computer Graphics (2013), ACM, pp. 107–114. <https://doi.org/10.1145/2508244.2508258>.
- [RHI*15] RIZZI S., HERELD M., INSLEY J., PAPKA M. E., URAM T., VISHWANATH V.: Large-scale parallel visualization of particle-based simulations using point sprites and level-of-detail. In Proc. EGPGV (2015), The Eurographics Association, pp. 1–10. <https://doi.org/10.2312/pgv.20151149>.
- [Ric77] RICHARDS F. M.: Areas, volumes, packing, and protein structure. *Annu. Rev. Biophys. Bioeng.* 6, 1 (1977), 151–176. <https://doi.org/10.1146/annurev.bb.06.060177.001055>.
- [SD] SCHRÖDINGER L., DELANO W.: PyMOL. URL: <http://www.pymol.org/pymol>.
- [SdG10] SEELIGER D., DE GROOT B. L.: Ligand docking and binding site analysis with PyMOL and Autodock/Vina. *J. Comput. Aid. Mol. Des.* 24, 5 (2010), 417–422. <https://doi.org/10.1007/s10822-010-9352-6>.
- [SKL*18] SKÅNBERG R., KÖNIG C., LINARES M., JÖNSSON D., NORMAN P., HOTZ I., YNNERMAN A.: VIA-MD: Visual Interactive Analysis of Molecular Dynamics. In *Proc. MolVA* (2018). <https://doi.org/10.2312/molva.20181102>.
- [SKPE19] SCHATZ K., KRONE M., PLEISS J., ERTL T.: Interactive visualization of biomolecules' dynamic and complex properties. *Eur. Phys. J. Spec. Top.* 227, 14 (2019), 1725–1739. <https://doi.org/10.1140/epjst/e2019-800162-y>.
- [SOS96] SANNER M. F., OLSON A. J., SPEHNER J.-C.: Reduced Surface: An efficient way to compute molecular surfaces. *Biopolymers* 38, 3 (1996), 305–320. [https://doi.org/10.1002/\(SICI\)1097-0282\(199603\)38:3\(305::AID-BIP4\)3.0.CO;2-Y](https://doi.org/10.1002/(SICI)1097-0282(199603)38:3(305::AID-BIP4)3.0.CO;2-Y).
- [TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: can it facilitate? *Int. J. Hum-Comput. St.* 57, 4 (2002), 247–262. <https://doi.org/10.1006/ijhc.2002.1017>.
- [UHPJ94] UPPENBERG J., HANSEN M. T., PATKAR S., JONES T. A.: The sequence, crystal structure determination and refinement of two crystal forms of lipase B from *Candida antarctica*. *Structure* 2, 4 (1994), 293–308. [https://doi.org/10.1016/S0969-2126\(00\)00031-9](https://doi.org/10.1016/S0969-2126(00)00031-9).
- [VBJ*17] VAD V., BYŠKA J., JURČÍK A., VIOLA I., GRÖLLER E., HAUSER H., MARQUES S. M., DAMBORSKÝ J., KOZLÍKOVÁ B.: Watergate: Visual exploration of water trajectories in protein dynamics. In *Proc. EG VCBM* (2017), The Eurographics Association. <https://doi.org/10.2312/vcbm.20171235>.
- [VDLSH*05] VAN DER SPOEL D., LINDAHL E., HESS B., GROENHOF G., MARK A. E., BERENDSEN H. J. C.: GROMACS: Fast, flexible, and free. *J. Comput. Chem.* 26, 16 (2005), 1701–1718. <https://doi.org/10.1002/jcc.20291>.
- [vdW73] VAN DER WAALS J. D.: *Over de continuïteit van den gas-en vloeïstoofstand*. PhD thesis, Hoogeschool te Leiden, 1873.
- [VHG*18] VÁZQUEZ P.-P., HERMOSILLA P., GUALLAR V., ESTRADA J., VINACUA À.: Visual analysis of protein-ligand interactions. *Computer Graphics Forum* 37, 3 (2018), 391–402. <https://doi.org/10.1111/cgf.13428>.
- [WHLS19] WANG J., HAZARIKA S., LI C., SHEN H.: Visualization and visual analysis of ensemble data: A survey. *IEEE Transactions on Visualization and Computer Graphics* 25, 9 (2019), 2853–2872. <https://doi.org/10.1109/TVCG.2018.2853721>.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting info