Workshop on Molecular Graphics and Visual Analysis of Molecular Data (2018)
J. Byška, M. Krone, and B. Sommer (Editors)

Atomic Accessibility Radii for Molecular Dynamics Analysis

N. Lindow, D. Baum, and H.-C. Hege

Zuse Institute Berlin, Germany

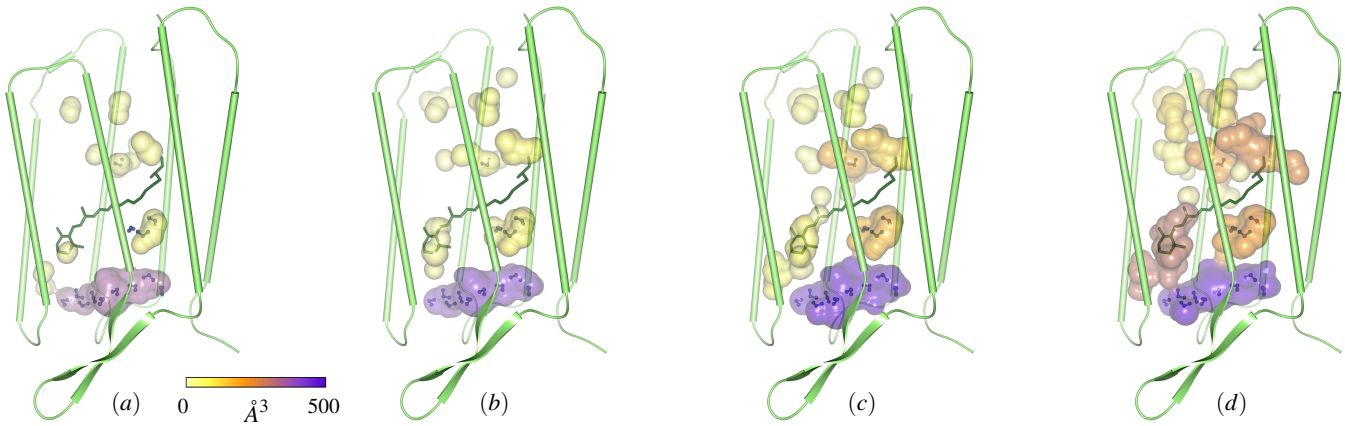


Figure 1: *Cavities and water molecules in a bacteriorhodopsin monomer. The cavities, which are depicted by surfaces colored according to their volume, were extracted for different atomic radii: (a) van der Waals radii [RT96], (b) individual atomic radii, (c) atom-type radii and (d) element radii. For (b) to (d), the radii were computed with the method proposed in this paper. Note that the water molecules depicted by ball-and-stick models are not fully contained in the cavities computed for vdW radii (a).*

Abstract

In molecular structure analysis and visualization, the molecule's atoms are often modeled as hard spheres parametrized by their positions and radii. While the atom positions result from experiments or molecular simulations, for the radii typically values are taken from literature. Most often, van der Waals (vdW) radii are used, for which diverse values exist. As a consequence, different visualization and analysis tools use different atomic radii, and the analyses are less objective than often believed. Furthermore, for the geometric accessibility analysis of molecular structures, vdW radii are not well suited. The reason is that during the molecular dynamics simulation, depending on the force field and the kinetic energy in the system, non-bonded atoms can come so close to each other that their vdW spheres intersect. In this paper, we introduce a new kind of atomic radius, called 'atomic accessibility radius', that better characterizes the accessibility of an atom in a given molecular trajectory. The new radii reflect the movement possibilities of atoms in the simulated physical system. They are computed by solving a linear program that maximizes the radii of the atoms under the constraint that non-bonded spheres do not intersect in the considered molecular trajectory. Using this data-driven approach, the actual accessibility of atoms can be visualized more precisely.

CCS Concepts

•***Computing methodologies*** → *Molecular simulation;* •***Applied computing*** → *Molecular structural biology;*

1. Introduction

Space-filling representations of molecular structures play an important role both for visualization and geometric analysis. In particular, the analysis of cavities, channels, pockets and binding sites is often based on molecular representations in which the molecule's atoms are simplified by hard spheres that are parametrized by their positions and radii. Due to the high repulsive force between close non-bonded atoms, two atoms cannot get too close to each other. Thus, to some extent and for specific purposes it is justified to consider atoms as hard spheres. For the visualization and geometric analysis of molecular structures, van der Waals (vdW) spheres are the most often used hard spheres. They are named after Johannes Diderik van der Waals, who was the first to recognize that each atom occupies a finite amount of space. One way to determine vdW radii is to analyze the attractive and repulsive interactions between non-

DOI: 10.2312/molva.20181101

bonded atoms. For distances between 3 and 4 Å, two non-bonded atoms start to attract each other. The attraction increases with decreasing atom distance. At a certain distance, the outer electron regions of both atoms start to overlap which results in a strong repulsive force [Str95]. This behavior is often physically modeled by the Lennard-Jones potential (Fig. 2). Using the Lennard-Jones potential, the vdW radii are often defined such that the sum of the vdW radii is equal to the vdW contact distance. Due to the thermodynamical fluctuations, the distance between two atoms is not fixed. As a result, the distance can become smaller than the vdW contact distance so that vdW spheres of non-bonded atoms can intersect.

Although vdW spheres are a useful representation of atoms for many applications, they must be handled with care when studying the accessibility of a molecule's atoms [LN98]. One reason is that vdW spheres of non-bonded atoms might intersect, which contradicts our intuition of accessibility. Another problem is that several definitions of vdW radii exist in the literature. Thus, different molecular visualization and analysis tools use different radii. Hence, the results are less objective than commonly believed. Furthermore, vdW radii are generally defined per chemical element. This ignores that attracting and repulsive forces might also depend on the number and types of covalently bonded atoms [LN98]. Finally, and this is true for all fixed atom radii, vdW radii do neither take into account the particular molecular system at hand nor its thermodynamical state. While for static molecular structures the latter problem cannot be easily circumvented, this is possible for molecular systems simulated using molecular dynamics (MD).

In most MD simulations, the Lennard-Jones potential is parametrized depending on pairs of atom types, where an atom type is described by the chemical element and the covalently bonded atoms. Hence, MD trajectories represent rich information, in particular about physically modeled atomic interactions but also about the thermodynamical state of the system. All of this information is represented by the interatomic distances occurring in the trajectory. For example, if the system is simulated at a high temperature, the interatomic distances will vary more strongly and the minimal interatomic distances will be smaller than for a lower temperature.

In this paper, we introduce a new type of atomic radius based on the analysis of MD trajectories. We call this radius *atomic accessibility radius* or simply *accessibility radius*. The name of the radius suggests that it is designed to better reflect the accessibility. Thus it is mainly suited for the geometric accessibility analysis of a molecule's atoms as well as for molecular visualization with the purpose of revealing the accessibility of certain regions of the molecule. We derive the atomic accessibility radii utilizing a data-driven approach that analyzes the minimal interatomic distances of non-bonded atoms of an MD trajectory. Note that for each molecular system and each simulation trajectory, the radii might differ, thus taking into account the specificity of the particular simulation. One property of the atomic accessibility radii is that all non-bonded atoms do not intersect throughout the whole MD trajectory. We guarantee this by optimizing a linear program where the non-intersection is formulated as constraints and the radii are maximized. In summary, we see as main contributions of this work

- an efficient algorithm to compute all minimal distances between non-bonded atoms in an MD simulation;
- a linear program to compute maximal accessibility radii with the constraints that non-bonded atoms do not intersect;
- a technique to compute accessibility radii for different abstraction levels (elements, atom types, individual atoms);
- an increase of the awareness that atomic radii are not universal constants but are defined with regard to a particular purpose; molecular visualizations based on atomic radii therefore are less objective than commonly thought and specifically vdW radii do often not correctly reflect the accessibility of atoms.

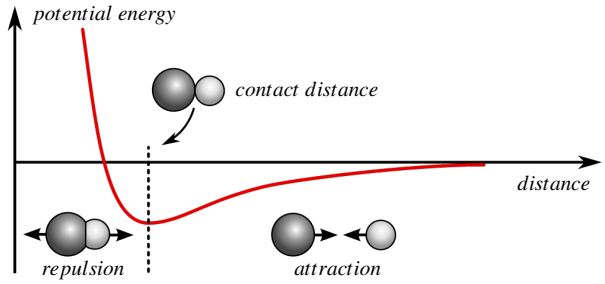


Figure 2: *Illustration of the vdW forces between hydrogen and carbon. The forces are modeled by the Lennard-Jones potential.*

2. Related Work

Since the PhD thesis of van der Waals (1873), postulating atoms and molecules of finite size to establish a correct equation of state for gases and liquids, there is a long history on determining atomic radii. Landmarks are the work of Bragg [Bra20], presenting radii for major chemical elements (and even visualizing crystal structures, composed of atoms depicted as spheres), and the book of Pauling [Pau39], giving a general account of different types of atomic radii in bonded (covalent, metallic, and ionic) and non-bonded states. Slater [Sla64] presented improved sets of empirical atomic radii, set up such that the sum of the radii of two atoms forming a bond in a crystal or molecule gives an approximate value of the internuclear distance. Intermolecular van der Waals (vdW) radii of the nonmetallic elements have been assembled into a list of recommended values for volume calculations by Bondi [Bon64]. Various methods have been devised – mechanical, crystallographical, electrical, optical and computational – to empirically determine vdW radii. They give similar, yet different values. For physical reasons why vdW radii vary with the particular chemical environment, see e.g. Bondi [Bon64]. Rowland et al. [RT96] presented new results considering the non-bonded contact distances in organic crystals. They recognized that the radii defined by Bondi are quite similar to their results and only differ for a few elements. Li and Nussinov [LN98] noted that polar atoms usually need more space when neighbored to other polar atoms. They derive Coulombic radii from the contact with water and suggest using these radii for polar atoms instead of vdW radii, in particular for cavity analysis and docking computations. An extensive overview of vdW radii of elements was given by Batsanov [Bat01]. In a more recent work Batsanov [Bat11] extended this work by atomic radii for metals, derived from empirical data and an equation of state for solids.

Based on the vdW spheres or other hard sphere models, several space-filling molecular representations have been defined. Among these are the vdW surface, the solvent accessible surface (SAS) [LR71], the solvent excluded surface (SES) [Ric77, C*83],

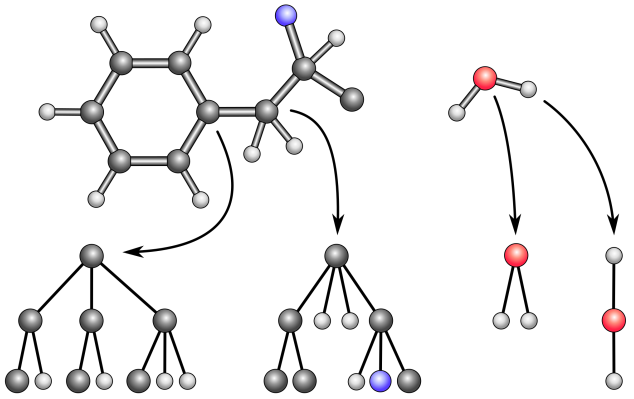


Figure 3: *Illustration of different atom type trees with $h_{max} = 2$. The colors represent the chemical elements, where hydrogen is white, carbon is gray, nitrogen is blue, and oxygen is red.*

the Gaussian molecular surfaces [GP95], and other molecular surfaces [Whi12, KKF*17]. How well these representations describe accessibility strongly depends on the radii of the used spheres.

Apart from visualization, which is often a first means to analyze the accessibility of a molecule's atoms, a more explicit means is cavity analysis. Most cavity detection algorithms are based on space-filling molecular representations. Especially the vdW surface or models based on the vdW surface, like the SES or Gaussian surfaces, are often used to visualize and extract cavities and channels in proteins. Hence, these structures directly depend on the used atomic radii. A recent review article [KKL*16] gives a good and detailed overview of the cavity analysis methods and tools, all of which are influenced by the choice of atomic radii.

3. Method

The general idea of our method is to derive the atomic accessibility radii from the minimal distances between non-bonded atoms of a given MD trajectory. The derived radii should be maximal such that the intersection of all pairs of non-bonded atoms is empty. This leads to an optimization problem, where the objective function is the sum of a set of accessibility radii, which is to be maximized under the constraints that no pair of non-bonded atom spheres intersects when using the derived radii as sphere radii. In the following we present the method to derive accessibility radii from an MD simulation in detail, describing three levels of abstractions.

3.1. Definition of Atom Types

The method we propose can be used to compute accessibility radii for chemical elements, for specific atom types, or for individual atoms. While the chemical element is always given as input, there exist different definitions for the atom type depending on the neighborhood of an atom in terms of covalent bonds and other properties. The type of an atom is more specific than its chemical element type. For example, two hydrogen atoms of a water molecule are clearly of the same atom type, but they are different to a hydrogen atom connected by a covalent bond to a carbon atom. Simulation programs parameterize the atom type differently. Sometimes the atom types are given by specific atom names in the data, as, for example, by the CHARMM simulation program [BBM*09]. To overcome

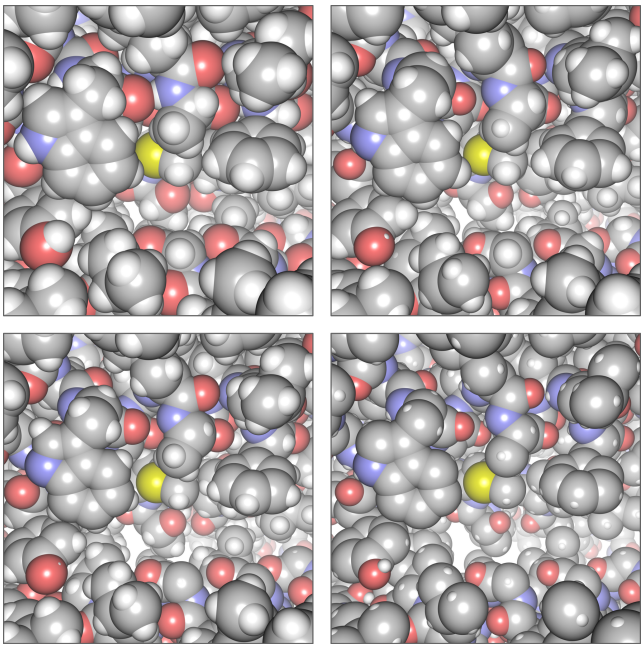
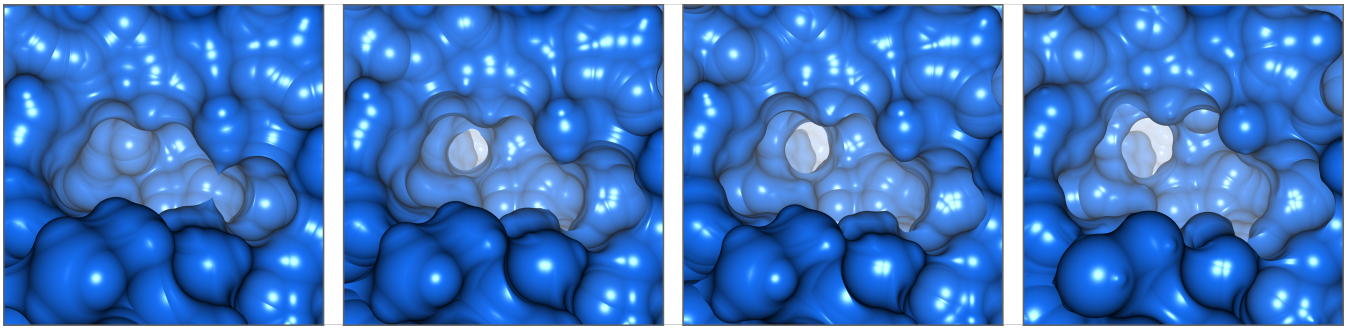


Figure 4: *Space-filling models with different atomic radii. The upper images show vdW radii [RT96] (left) and computed accessibility radii for individual atoms (right). The lower images show the computed radii for atom types (left) and chemical elements (right). The atoms are colored according to the CPK schema.*

the problem that different atom types are used by different simulation programs, we propose a systematic definition of atom types that is based on the covalent bonds and the chemical elements in the neighborhood.

In order to give such a formal definition, it is most convenient to consider a molecule with n atoms as a graph $G = (V, E)$ where the set of nodes V represents the atoms, and the set of edges E denotes the covalent bonds between the atoms. Let v_i and v_j represent two atoms i and j of the molecule. Then, $e = (v_i, v_j) \in E$ if and only if i and j are connected by a covalent bond. Let $h(i, j)$ be the bond distance between two atoms i and j. It is defined as the number of bonds of the shortest path in G from v_i to v_j. Now, given the graph G and a maximal bond distance $h_{max} \in \mathbb{N}$, for each atom i, we can extract a tree $T_i(h_{max})$ that contains all nodes $v_j \in V$ with $h(i, j) \leq h_{max}$. The tree is rooted at v_i and constructed as follows: The children of v_i are all nodes v_j with $h(i, j) = 1$. More general, the child nodes of a node v_k are all nodes v_l with $h(k, l) = 1$ and $h(i, l) = h(i, k) + 1$ under the condition that $h(i, l) \leq h_{max}$. The tree T_i of an atom i represents its atom type. Therefore, in the following we call it 'atom type tree'. Figure 3 shows atom type trees for four different cases. Two atoms i and j have the same atom type if their corresponding atom type trees T_i and T_j are isomorphic. Efficient algorithms to determine whether two trees are isomorphic are, for example, described by Valiente [Val02].

The maximal bond distance for the creation of the atom type trees can be specified by the user, but in most cases a maximal bond distance of 2 seems appropriate and is similar to what is considered by simulation programs.



(a) (b) (c) (d)

Figure 5: *Close-up of the solvent excluded surface showing a cavity in bacteriorhodopsin. The surface is computed using vdW radii [RT96] (a) as well as accessibility radii computed for individual atoms (b), atom type (c), and chemical elements (d).*

3.2. Non-Bonded Atoms

For calculating accessibility radii as proposed in this work, we consider the Euclidean distances between atoms that are not bonded to each other. The term 'non-bonded' requires a clear definition that is given in this paragraph.

Molecular dynamics simulation programs handle atoms connected by covalent bonds differently than atoms that are not connected by covalent bonds. Non-bonded forces, i.e. Coulomb and vdW interactions, are only applied between pairs of atoms that are either not at all covalently connected, or separated by more than three covalent bonds. Bonded forces, on the other hand, are modeled by harmonic springs for bond stretching and bond angle bending motions as well as bond twisting motions. Due to these forces, bonded atoms can get closer to each other than atoms interacting by non-bonded forces. The maximal number of atoms involved in a bonded interaction occurs for bond twisting motions. A twisting motion includes 4 atoms connected in a chain by 3 bonds. Therefore, we define a pair of atoms i and j as non-bonded if there is no covalent connection over three or less bonds, that is if $h(i,j) > 3$.

Whether or not two atoms are non-bonded can be tested very quickly in the following way. First of all, we compute for each atom a list of atoms that are connected directly or indirectly via at most three covalent bonds. We call such a list blacklist, where $B_i \subset \mathbb{N}$ is the blacklist of atom i. In classical MD simulations, covalent bonds are fixed over the entire simulation. Thus, we only have to compute the blacklist for each atom once. This can be efficiently done on the molecular graph G defined in Sec. 3.1. To do so, we use Dijkstra's shortest path algorithm to compute all atoms j with $h(i,j) \leq 3$. Afterwards we use the quicksort algorithm to sort the atom indices of each blacklist. This allows us to quickly check with a binary search if another atom is non-bonded or not.

3.3. Distance Computation

Since the calculation of accessibility radii is based on the computation of minimal Euclidean distances between pairs of non-bonded atoms, here we describe how these distances are computed.

As mentioned before, our approach works for element types, atom types, and individual atoms. To generalize the description of the method, we first introduce an atom classifier $c : \{1,...,n\} \longrightarrow \{1,...,m\}$ that maps each atom to a class. Depending on whether we want to compute the accessibility radii of element types, atom types or individual atoms, a class represents a chemical element, an atom type, or an individual atom, respectively. Hence, for individual accessibility radii, c is the identity map and $m = n$.

Let $D \in \mathbb{R}^{m,m}$ be a distance matrix, storing the minimal distances between the atom classes. Note that D is symmetric and that we initialize it with a user-defined distance cutoff value d_{max}. For each time step and each atom i, we collect all neighboring atoms $N_i \subset \mathbb{N}$ of i. Therefore the neighborhood of atom i is defined as the set of all atoms whose distance to i is smaller than d_{max}. Note that i is not a neighbor of itself, so $i \notin N_i$. In addition, we have to consider the boundary conditions of the simulation. A fast neighborhood detection can be achieved using a 3-dimensional grid data structure to store the atom positions. For each neighbor $j \in N_i$, we check if j is stored in the blacklist B_i of atom i. Since B_i is sorted, this can be done in an efficient way using a binary search. If $j \in B_i$, we can ignore it; otherwise, we compute the distance d between the atoms i and j. If $d < d_{k,l}$, where $k = c(i)$ and $l = c(j)$, we have to update the minimal distance by setting $d_{k,l}$ and $d_{l,k}$ to d. After all time steps and atoms have been processed, D contains all minimal distances between the defined atom classes that appear in the molecular trajectory.

3.4. Radii Computation

In order to compute maximal radii for the atom classes, we have to solve an optimization problem. The optimization problem can be formally described by a set of variables, an objective function, a set of constraints, and, optionally, a set of bounds. The variables represent the radii of different atom classes. Our goal is to maximize all of them, so the objective function to be maximized is the sum of all radii. The constraints are given by the minimal distances between the atom classes. We want to avoid that non-bonded atoms intersect, so each constraint describes an inequality where the sum of two radii should be smaller than or equal to the minimal distance of the corresponding atoms. Optionally, one can add further bounds for the variables. A lower bound for the radii is 0, because accessibility radii should always be non-negative. An upper bound is not necessary, but one could optionally use the vdW radii suggested by Rowland et al. [RT96] or even larger values. To compute the atom class radii, we solve the following problem:

$$\begin{aligned} \textit{maximize} \quad & \sum_{i=1}^{m} w_i \cdot r_i \\ \textit{constraints} \quad & r_i + r_j \leq d_{i,j} \quad i \leq j | i,j \in \{1,...,m\} \\ \textit{bounds} \quad & r_i \geq 0. \end{aligned}$$

Here, r_i are the radii of the atom classes weighted by $w_i \in \mathbb{R}$. Except for the minimal distances between non-bonded atoms, the problem has no further information about the physics of the simulation. In particular, it has no information about the different elements. In case of water, for example, this can lead to hydrogen atoms that are larger than oxygen atoms. The reason for this is that the non-bonded interactions in water mainly take place between hydrogen atoms and oxygen atoms. A hydrogen atom tends to get closer to the oxygen atom of another water molecule than to the hydrogen atoms. Because, there are twice as many hydrogen atoms as oxygen atoms in water, the optimal solution would be to generate as large as possible radii for the hydrogen atoms. This is mainly due to the fact that it has not been specified how to distribute the minimal distances to the corresponding accessibility radii. We add this information by introducing relationships between the sizes of the accessibility radii, which is realized by weights w_i for classes of atoms.

Assuming a spherical shape and a constant density of bonded atoms, an accessibility radius is proportional to $\sqrt[3]{m_e}$ of the mass m_e of the element. To ensure that each atom and not only the class is represented in the optimization, we need to take into account the number of atoms per class c_i. Hence, we weight each radius by $w_i = \sqrt[3]{m_e} \cdot c_i$. This creates a relationship between the radii which only leads to a tendency not a strict relationship in the solution. This particular weighting is a physical approximation but it can also be replaced by other application-dependent relationships.

Since the objective function of this problem is linear with respect to the variables and the constraints, and the variables have a lower bound of 0, the optimization problem belongs to the linear optimization problems, also known as linear programs (LPs). One can solve such a classical continuous linear program using a simplex-algorithm or an interior point method [Mur83]. An open source implementation of a revised simplex algorithm is SOPLEX [Wun96] integrated in the SCIP [Ach09] framework. Two of the fastest commercial LP solvers are CPLEX [CPL] and Gurobi [Gur].

3.5. Individual Accessibility Radii

Classical molecular dynamics simulations of biomolecules usually contain hundreds of thousands up to several millions of atoms. For the computation of individual accessibility radii, this can lead to very large distance matrices that possibly do not fit into the main memory. Furthermore, the number of constraints $m(m+1)/2$ is quadratic in the number of atoms m. Even the fastest LP solvers, for example, Gurobi and CPLEX, have problems to solve such large problems. To reduce the size of the LP, we can reduce the number of constraints by not storing the minimal distances to all atoms. This is possible for two reasons. First, not all atoms in the blacklist need to be considered and, thus, we do not need to store the distances to them. Second, atoms whose distance is always larger than d_{max} also do not need to be stored.

However, this reduction might still not suffice, since MD simulations typically contain a lot of water molecules and these water molecules move very fast. Hence, many atoms in the simulation have minimal distances smaller than d_{max} to a lot of atoms, namely the hydrogen and oxygen atoms of the water. If this is the case, we use a heuristic that restricts the number of minimal distances for each atom to a fixed number k. That is, for each atom, we store maximally k minimal distances. Note that in this case, each distance needs to be labeled with the corresponding atom, because we do not have a distance matrix D anymore but only lists of distances. Instead of storing all minimal distances to atom i, we only store the minimal distances to the atoms that come closest to atom i. If we applied this strategy, we would typically end up with distances to hydrogen atoms only, because the distances to hydrogen atoms are generally smaller than the distances to other elements. Therefore, we partition the k possible minimal distances into slots k_l for all occurring elements l, with $\sum k_l = k$. The slots are chosen proportional to the number of atoms of each element. For biomolecular simulations, this results in a large slot for hydrogen atoms, medium slots for carbon and oxygen, and very small slots for rare elements like nitrogen, sulfur, and phosphor. In each slot, we store only the smallest minimal distances according to the slot element. Hence, instead of a quadratic number of constraints, the overall number is now restricted to $k \cdot m$. In practice, we set k to a value between 50 and 100.

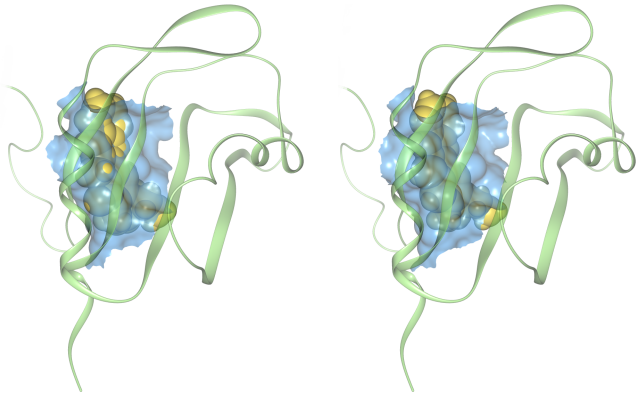


Figure 6: *Solvent excluded surface (blue) of a molecular pocket containing a ligand (trajectory 1 in Table 1). The surface was generated using vdW radii [RT96] (left) and computed individual accessibility radii (right).*

4. Results

In the following, we give computation timings as well as qualitative and quantitative results concerning the computation of the accessibility radii using different parameters.

All trajectories used for our radii computation were created by our cooperation partners using two different simulation programs and force fields. The first 3 trajectories in Table 1 were simulated using Gromacs [PPS*13] with the Amber force field [CDC*12]. For the other trajectories, NAMD [PZKK02] and the CHARMM force field [BBM*09] were used. All trajectories contain between 200.000 and 300.000 atoms. The number of time steps (#TS) used for the computation of the radii is given in Table 1.

For the blacklist computation, we used a bond distance of 3, that is, only atoms with a minimal bond distance larger than 3 are considered as non-bonded. This corresponds to the modeling of bonding forces in MD simulations.

To give an impression for the amount of intersections of non-bonded atoms within a trajectory, we computed the average distribution of the intersection overlap for the vdW radii by Rowland et al. [RT96] (Fig. 7). For the first two trajectories ∼96 % of the atoms

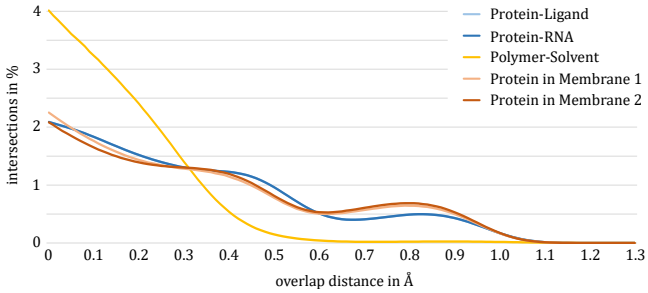


Figure 7: *Average distribution of the pairwise intersections of non-bonded atoms per time step depending on the overlap distances for the five data sets in Table 1 when using the vdW radii proposed by Rowland et al. [RT96].*

are involved in intersections per time step. For the polymer-solvent trajectory still ∼50 % of the atoms intersect and within the protein membrane trajectories ∼76 % and ∼82 % of the atoms intersect.

4.1. Performance

In this section, we give timings for the computation of the accessibility radii for an MD trajectory with approximately 250.000 atoms. All computations were executed on a desktop system (Intel Xeon X5650, 2.66 GHz, 24 GB RAM).

All pre-processing steps of our algorithm scale linearly with the number of atoms. For the computation of the blacklists of all atoms, approximately 1 minute was needed. Note that the blacklists need to be computed only once for the whole trajectory. No optimizations were carried out for this part of the algorithm. The second pre-processing step in our algorithm is the computation of all minimal distances. For each time step, this took approximately $1-2$ seconds. The modified distance computation, described in Section 3.5, for up to 100 minimal distances is in the same time scale.

Linear programs (LPs) can be solved in polynomial time, but it is difficult to give an exact performance analysis for a specific kind of problem. Often, the time to solve an LP depends mainly on the number of variables and constraints. For the computation of the radii for elements or atom types, the LPs consist only of up to 150 variables and a few thousand constraints. These LPs can be solved within less than a second by most LP solvers, like SOPLEX [Wun96]. In contrast, the computation of individual accessibility radii is much more expensive. SOPLEX was not able to compute the optimal solution within 3 days. For 50 minimal distances per atom, Gurobi [Gur] took approximately 5 minutes to solve the problem. When using 100 minimal distances, this time significantly increased to approximately 45 minutes.

4.2. Radii for Elements

We computed accessibility radii per element for 5 molecular dynamics simulation trajectories. The results are shown in Table 1 together with the vdW radii proposed by Rowland et al. [RT96]. We listed only the values for hydrogen (H), carbon (C), nitrogen (N), oxygen (O), and sulfur (S), since these were the element types that were present in the trajectories. As expected, nearly all computed accessibility radii are smaller than the vdW radii proposed by Rowland et al. [RT96]. Especially the computed radii for the hydrogen atoms are much smaller. Another observation we can make is that the differences in the element radii between different trajectories are rather small. The largest radius difference is 0.23 $\mathring{A}$ for hydrogen.

4.3. Radii for Atom Types

For computing the atom type radii, we used a maximal tree depth of 2, so $h_{max} = 2$. This results in 109-133 different atom types for the trajectories 1, 2, 4, and 5. The radii for these four trajectories are given in the supplementary material. Trajectory 3 is a quite different system that contains only 50 atom types and is therefore excluded from the comparison. Altogether 180 different atom types occurred in the four molecular systems, with quite similar radii for each atom type. Only for one atom type, the maximal radius difference was larger than 0.3 Å, and for most atom types, the maximal radius difference was below 0.15 Å. Additionally, for trajectory 1, the atom type radii and the element radii are shown in Figure 8, left. Here, the atom types are grouped by their tree root elements. The atom type radii are depicted as filled circles along the x-axis, while the element radii are depicted as black crosses.

It can be observed that the radii for hydrogen atoms range from approximately 0.25 $\mathring{A}$ to approximately 0.9 $\mathring{A}$. While the hydrogens with the smallest radii are those from water, the largest hydrogen radii were computed for hydrogens that are connected to a carbon atom. The radii for most hydrogen atom types vary between 0.75 $\mathring{A}$ and 0.8 $\mathring{A}$. One can see that the element radius of hydrogen is smaller than the radii computed for most of the hydrogen atom types. This can be explained by the fact that the simulation trajectories contain many water molecules whose hydrogen atoms move closer to other atoms than hydrogen atoms of the protein or the membrane. Since we use only the minimal distances between elements, they mainly come from the water molecules. Larger minimal distances, for example, inside the membrane or inside the protein, are not considered.

The usage of radii for atom types allows us to select more than one radius per element. Thus, the optimizer is able to use much larger radii for many hydrogen atoms. This also effects the radii of the carbon, nitrogen, and oxygen types, where a similar tendency can be seen. Due to the larger radii of these elements, rare elements like sulfur or phosphor become a bit smaller.

4.4. Radii for Individual Atoms

The computation of accessibility radii for individual atoms was carried out for all trajectories in Table 1. The results can be visually

Table 1: *Comparison of the vdW radii proposed by Rowland et al. [RT96] and the computed element radii for 5 molecular dynamics trajectories with the given number of time steps (#TS). The radii are given in Å.*

Radii Source	#TS	H	C	N	O	S
Rowland [RT96]	-	1.10	1.77	1.64	1.58	1.81
Protein-Ligand	4000	0.23	1.42	1.21	1.16	1.58
Protein-RNA	4000	0.22	1.41	1.25	1.15	1.64
Polymer-Solvent	13000	0.46	1.36	1.37	1.01	-
Protein in Mem.	350	0.44	1.46	1.31	1.08	1.60
Protein in Mem.	2550	0.34	1.40	1.33	1.03	1.70

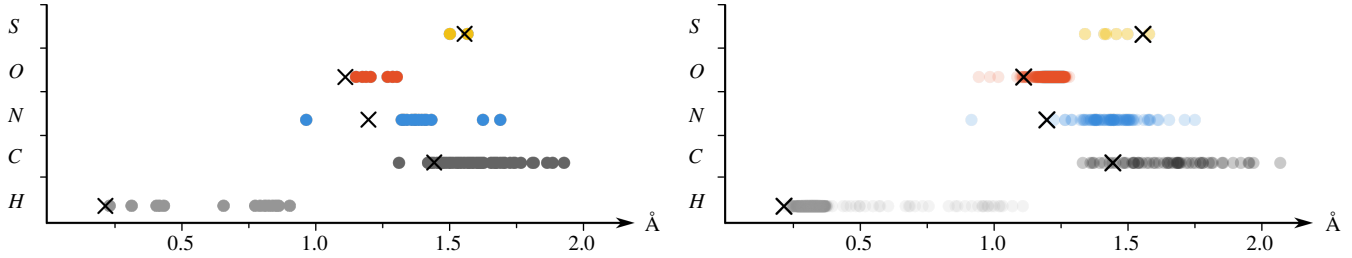


Figure 8: *Radii values for different atom types (left) and individual atoms (right) compared to the radii computed for each chemical element (black crosses). The radii of the atom types and individual atoms (colored circles) are grouped according to their chemical element.*

compared directly in Figure 4 or indirectly by the internal cavities and molecular representations in Figures 1, 5, 9, and 6. Similar to the atom type radii, we plotted the radii of the protein-ligand trajectory as circles in a diagram in Figure 8, right. The radii were grouped again by the corresponding chemical element. In order to show the distribution of the radii, we blended the circles representing the radii with a constant transparency value. We observed that the radii are similar to the radii for atom types, but that they tend to be a bit larger. Of course, this results in less and smaller cavities.

4.5. Radii Comparison and Impact

Since our radii computations are based on the minimal distances between non-bonded atoms, it is obvious that the radii are in general smaller than most presented van der Walls radii, whose purpose is to reproduce the volume [Bon64] or the contact distance of the Lennard-Jones potentials [RT96] (Fig. 2). Furthermore, one can observe that the more specific the radii definitions are, the larger are the accessibility radii. Hence, in general, radii for chemical elements are smaller than radii for atom types which are again smaller than radii for individual atoms (Fig. 4). This has a direct impact on the results of all surface visualization models based on hard atom spheres. One example is the visualization of the solvent excluded surface (SES), which shows all accessible regions of a molecule for a user-defined probe sphere. A close-up of the surface for a bacteriorhodopsin protein with a probe sphere radius of 1.4 $\AA$ is shown in Figure 5. One can clearly see that the size of the cavity in the middle of the image depends on the atomic radii. For classical vdW radii, the cavity is much smaller compared to our computed radii. Thus, the probe sphere can access deeper regions in the protein. Figure 6 shows another example of the SES of the back side of a molecular pocket. The pocket includes a ligand bound to the protein. By using vdW radii, the ligand intersects the surface of the protein. This is not the case with our computed individual accessibility radii.

Analogously, the radii also have a direct impact on the results of feature extraction algorithms, like the molecular cavity detection. In Figure 1, all cavities in a bacteriorhodopsin monomer are shown based on the 4 different atomic radii. The cavity detection was done by the method presented by Lindow et al. [LBH11]. Again one can see that more and larger cavities are detected for our computed radii than for the vdW radii. One can also observe that for vdW radii, the cavities are too small to completely enclose the included water (Fig. 9) or that cavities are even missing (Fig. 1).

5. Discussion and Conclusion

For most molecular visualizations and analyses based on hard atom spheres, vdW radii are used. However, in the literature, different van der Waals radii can be found whose utilization leads to different visualization and analysis results. This poses a problem, since the results of geometric analyses are not as objective as they may seem. Furthermore, vdW radii are computed with the purpose to reproduce the volume of the molecules [Bon64] or to approximate the equilibrium distance of the Lennard-Jones potentials [RT96]. For this reason, vdW radii are not suitable to model the accessibility by hard spheres, because the minimal distance between two non-bonded atoms is smaller than the sum of their vdW radii. This might lead to problems, e.g. when using the radii for visualization models showing accessibility or for extraction of geometric structures like cavities.

To overcome this problem, we have proposed a method to compute atomic accessibility radii directly from an MD trajectory. The computation is based on the minimal distances between pairs of non-bonded atoms. Our method guarantees that the intersection of the atom spheres of any two non-bonded atoms is always empty. This results in generally smaller atomic radii compared to the vdW radii. Thus, our computed radii are better suited for the task of analyzing the accessibility in molecules, which is demonstrated in Figures 1, 6, and 9. Furthermore, the radii we compute are based on the physical properties considered in the simulation and, thus, they take into account the thermodynamical parameters of the simulation.

Since our method guarantees that non-bonded atoms do not intersect, we cannot miss geometrically accessible regions. This may lead to an overestimation of accessibility. However, this overestimation cannot be correctly handled by increasing the atomic radii, since our radii are maximal with respect to non-intersection of non-bonded atoms. Physical analysis methods are needed to finally determine which regions are really accessible and which are not.

Using the mass of the chemical elements for the weighting of the accessibility radii in the objective function of the LP is a physical approximation to generate relationships between the radii. Depending on the input data and the analyses task, other weightings might be more appropriate. However, for the analyses of the biomolecular data in this work, the weightings proved to be very suitable.

Apart from the weights, the accessibility radii computation depends solely on the distance analysis of non-bonded atoms. For a correct result, it is important that all necessary minimal distances are represented by the trajectory, otherwise the radii will be larger. This requires that the density of atoms is high enough to reflect the non-bonded minimal distances. In the case of biomolecular simulations, this density requirement is generally fulfilled. Hence the result mainly depends on the number of time steps of the trajectory.

The more time steps can be analyzed, the more accurate are the minimal distances.

One observation we made is that during computation of a radius value for each chemical element, the distances in D converge faster with the number of time steps to the minimal distances than during computation of radii for atom types. This can be explained by the fact that for the calculation of an element radius more statistical information from a single time step can be used than for the calculation of an atom type radius. This is even more the case for individual atoms. Overall, we can observe that the more specific the definition of the radii, the more time steps are required to get accurate results. Especially for individual accessibility radii, this requires several hundreds or thousands of time steps.

Note that the non-intersection of pairs of non-bonded atom spheres is not guaranteed for the computation of individual accessibility radii using a restricted number of minimal distances. One possibility to avoid intersections is to run the algorithm several times using the current distances of intersections as constraints and the radii of the previous step as upper bounds. However, in our results we did not observe such intersections.

Since our method to compute accessibility radii is based purely on information contained in the molecular trajectory, we do not need to explicitly consider parameters of the simulation program and the simulation run. Instead, we can compute accessibility radii from any molecular dynamics trajectory no matter what parameters were used. For a higher simulation temperature, for example, the accessibility radii might become smaller than for a lower temperature, thereby reflecting the higher energy of the molecule's atoms. This has a direct impact on the visualization models and the further geometric analysis tasks, like cavity detection.

Nearly all simulation programs use different Lennard-Jones potentials for different pairs of atom types. Especially with the computed radii for atom types and individual atoms, we are the first to represent and visualize these properties. For further cavity analysis tasks we suggest to use the radii computed for different atom types. Since most simulation programs also use specific behaviors for different atom types, the radii more accurately represent these properties than radii for element types. Furthermore, the radii computation for atom types does not require so many simulation time steps like the computation for individual atoms and the LP can be solved very fast even for very large data sets. The definition of the atom types by the covalently bonded neighborhood for $h_{max} = 2$ results in similar atom types than the types used in most simulation programs. However, the user can select other values for h_{max}. While increasing h_{max} leads to more specific radii and large LPs, decreasing h_{max} results in more general radii. For the case of $h_{max} = 0$, the atom types are equivalent to the chemical elements.

Our approach is of particular interest for the visualization of models showing possible accessible regions for other atoms, ions or substrates. One example is the SES (Fig. 5), which reflects better the possible accessibility using the atomic accessibility radii compared to the use of vdW radii. Furthermore, algorithms related to void detection will also benefit from using accessibility radii.

6. Future Work

We plan to evaluate our method in more detail by analyzing more cavity structures from simulation data. Specifically, we want to compare the accessibility of substrates using our method with the accessibility resulting from simulations. However, many processes like, for example, proton transport by a proton pump or docking processes of substrates are still difficult to simulate. Concerning the accessibility of water molecules, it would be particularly interesting to compare our results to the Coulombic radii for polar atoms [LN98]. We also plan to publish our atom-type radii after doing further comparisons with other MD trajectories. Additionally, we would like to compare the calculated accessibility using atomic accessibility radii with experimentally determined accessibility. However, it is not clear, how the latter can be achieved.

Furthermore, we plan to optimize our implementations to reduce the amount of time for the computation of the radii. Most of the pre-processing can be done in parallel. Thus, it is highly suitable for a GPU implementation. With such optimizations, solving the LP will become the most critical part of the whole pipeline even for thousands of time steps.

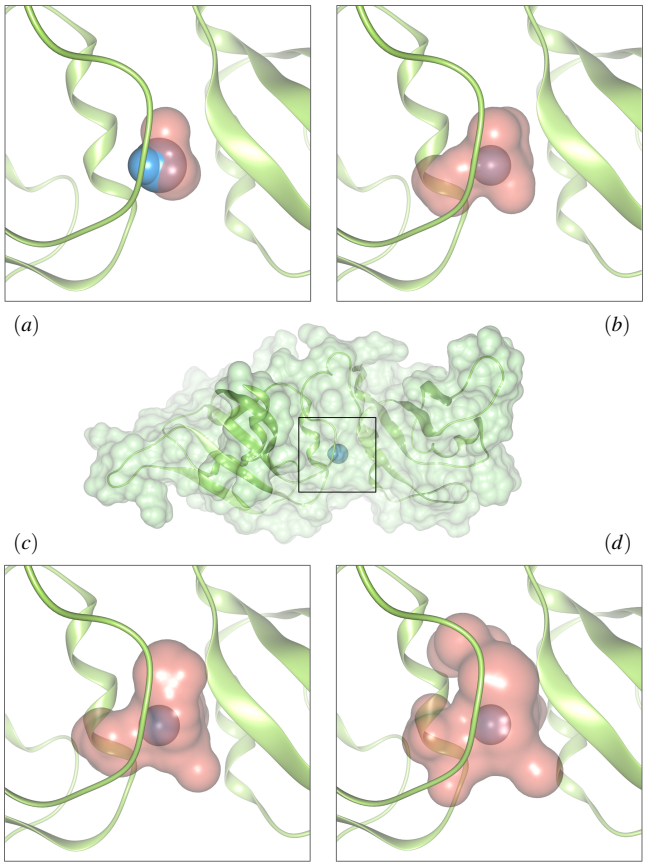


Figure 9: *Cavity containing a water molecule. The cavity was extracted using (a) vdW radii [RT96], (b) individual radii, (c) atom type radii, and (d) element radii. The middle image shows the complete protein (trajectory 2 in Table 1) with the location of the water.*

Acknowledgments

The authors would like to thank A.-N. Bondar from Freie Universität Berlin for providing the two protein-in-membrane molecular dynamic trajectories, and V. Durmaz from Zuse Institute Berlin for providing the protein-ligand and protein-RNA trajectories.

References

[Ach09] ACHTERBERG T.: SCIP: Solving constraint integer programs. *Math. Program. Comput. 1*, 1 (July 2009), 1–41. 5

[Bat01] BATSANOV S. S.: Van der Waals radii of elements. *Inorg. Mat. 37*, 9 (2001), 871–885. 2

[Bat11] BATSANOV S. S.: Thermodynamic determination of van der Waals radii of metals. *J. Mol. Struct. 990*, 1 (2011), 63–66. 2

[BBM*09] BROOKS B. R., BROOKS C. L., MACKERELL A. D., NILSSON L., PETRELLA R. J., ROUX B., WON Y., ARCHONTIS G., BARTELS C., BORESCH S., CAFLISCH A., CAVES L., CUI Q., DINNER A. R., FEIG M., FISCHER S., GAO J., HODOSCEK M., IM W., KUCZERA K., LAZARIDIS T., MA J., OVCHINNIKOV V., PACI E., PASTOR R. W., POST C. B., PU J. Z., SCHAEFER M., TIDOR B., VENABLE R. M., WOODCOCK H. L., WU X., YANG W., YORK D. M., KARPLUS M.: CHARMM: The biomolecular simulation program. *J. Comput. Chem. 30*, 10 (July 2009), 1545–1614. 3, 5

[Bon64] BONDI A.: van der Waals volumes and radii. *J. Phys. Chem. 68*, 3 (Mar. 1964), 441–451. 2, 7

[Bra20] BRAGG W. L.: The arrangement of atoms in crystals. *Lond. Edinb. Dubl. Phil. Mag. 40*, 236 (1920), 169–189. 2

[C*83] CONNOLLY M. L., ET AL.: Solvent-accessible surfaces of proteins and nucleic acids. *Science 221*, 4612 (1983), 709–713. 2

[CDC*12] CASE D. A., DARDEN T. A., CHEATHAM T. E., SIMMERLING C. L., WANG J., DUKE R. E., LUO R., WALKER R. C., ZHANG W., MERZ K. M., ROBERTS B., HAYIK S., ROITBERG A., SEABRA G., SWAILS J., GOETZ A. W., KOLOSSVÁRY I., WONG K. F., PAESANI F., VANICEK J., WOLF R. M., LIU J., WU X., BROZELL S. R., STEINBRECHER T., GOHLKE H., CAI Q., YE X., WANG J., HSIEH M. J., CUI G., ROE D. R., MATHEWS D. H., SEETIN M. G., SALOMON-FERRER R., SAGUI C., BABIN V., LUCHKO T., GUSAROV S., KOVALENKO A., KOLLMAN P. A.: Amber 12, 2012. 5

[CPL] ILOG CPLEX. http://www.ilog.com/products/cplex/. 5

[GP95] GRANT J. A., PICKUP B. T.: A Gaussian description of molecular shape. *J. Phys. Chem. 99*, 11 (1995), 3503–3510. 3

[Gur] Gurobi optimization. http://www.gurobi.com. 5, 6

[KKF*17] 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. *Comput. Graph. Forum 36*, 8 (2017), 178–204. 3

[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. *Comput. Graph. Forum 35*, 3 (2016), 527–551. 3

[LBH11] LINDOW N., BAUM D., HEGE H.-C.: Voronoi-based extraction and visualization of molecular paths. *IEEE Trans. Vis. Comput. Graphics 17*, 12 (2011), 2025 – 2034. 7

[LN98] LI A.-J., NUSSINOV R.: A set of van der waals and coulombic radii of protein atoms for molecular and solvent-accessible surface calculation, packing evaluation, and docking. *Proteins: Structure, Function, and Bioinformatics 32*, 1 (1998), 111–127. 2, 8

[LR71] LEE B., RICHARDS F. M.: The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol. 55*, 3 (1971), 379–400. 2

[Mur83] MURTY K. G.: *Linear Programming*. John Wiley & Sons Inc, New York, 1983. 5

[Pau39] PAULING L.: *The Nature of the Chemical Bond and the Structure of Molecules and Crystals: An Introduction to Modern Structural Chemistry*. 429 p., Cornell University Press, Ithaca, NY, 1939. 2

[PPS*13] PRONK S., PÃALL S., SCHULZ R., LARSSON P., BJELKMAR P., APOSTOLOV R., SHIRTS M. R., SMITH J. C., KASSON P. M., VAN DER SPOEL D., HESS B., LINDAHL E.: Gromacs 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics 29*, 7 (2013), 845–854. 5

[PZKK02] PHILLIPS J. C., ZHENG G., KUMAR S., KALE L. V.: NAMD: Biomolecular Simulation on Thousands of Processors. In *Proc. of Supercomputing, ACM/IEEE 2002 Conf.* (2002), pp. 1–18. 5

[Ric77] RICHARDS F. M.: Areas, volumes, packing, and protein structure. *Annu. Rev. Biophys. 6*, 1 (1977), 151–176. 2

[RT96] ROWLAND R. S., TAYLOR R.: Intermolecular nonbonded contact distances in organic crystal structures: Comparison with distances expected from van der Waals radii. *J. Phys. Chem. 100*, 18 (May 1996), 7384–7391. 1, 2, 3, 4, 5, 6, 7, 8

[Sla64] SLATER J. C.: Atomic radii in crystals. *J. Chem. Phys. 41* (1964), 3199. 2

[Str95] STRYER L.: *Biochemistry*. W.H. Freeman, 1995. 2

[Val02] VALIENTE G.: *Algorithms on Trees and Graphs*. Springer-Verlag, Berlin, 2002. 3

[Whi12] WHITLEY D.: Analysing molecular surface properties. In *Drug Design Strategies: Computational Techniquess and Applications*. Royal Society of Chemistry, 2012, pp. 184–209. 3

[Wun96] WUNDERLING R.: *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin, 1996. 5, 6