Decomposed Solvent Accessible Surface Area Calculations: Summary and Benchmark Calculations

Darian T. Yang[1] & Jeremy M. G. Leung[1]

[1]Department of Chemistry, University of Pittsburgh, Pittsburgh, Pennsylvania 15260, United States

---

Recently, we learned that when using CPPTRAJ for SASA calculations, especially for calculating partial or decomposed SASA contributions of your system (e.g. just helix 1 or only the alanine residues), there are inaccuracies that one should be aware of.

Summarizing this mailing list post (http://molecular-modeling.dq.ua.pt/wiki_files/SASA_LCPO_AMBER.pdf):

- The CPPTRAJ `surf` command (https://amberhub.chpc.utah.edu/surf/) calculates SASA using the LCPO method
    - This method was parameterized for total surface areas and is not expected to work well for decomposed SAs (even though it says it "supports").
- The CPPTRAJ `molsurf` command (https://amberhub.chpc.utah.edu/molsurf/) calculates the Connolly surface area (i.e. solvent-excluded surface) of the atom selection, but does not actually calculate the decomposed SA, and will just treat all other atoms as non-existent.
    - Thus if you calculate the SA of an alanine residue in one protein system vs another, the values will be relatively similar since it is only considering the alanine and no other atoms.
- Jason Swalis and Daniel Roe recommend not using either the `surf` or `molsurf` commands for decomposed SASA calculations, and to use VMD or Naccess.

Note that for calculations of the entire protein, these cpptraj commands should both work fine.

So what are the best alternatives for accurate, decomposed SASA calculations? Well, we ran some tests of various SASA implementations and confirmed that the LCPO method of CPPTRAJ can lead to inaccuracies and negative SASA values, especially for decomposed surfaces. MDTraj and freeSASA both obtain similar results, likely since they both use the Shrake and Rupley algorithm. VMD was between the CPPTRAJ LCPO and MDTraj/freeSASA values. Overall MDTraj, freeSASA, and VMD are all likely to provide reasonable decomposed SASA values which will be much more accurate than those from CPPTRAJ.

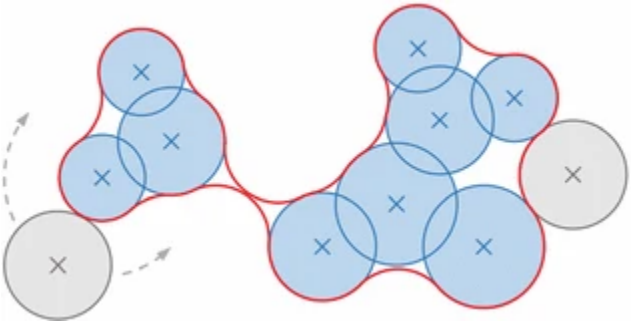
---

1. Introduction to protein surface calculation programs:

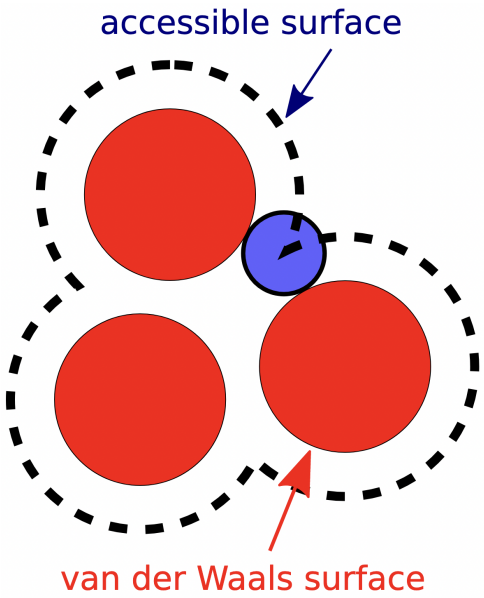
Connolly Surface (solvent-excluded surface) vs. Solvent Accessible Surface Area (SASA)

When we think of molecular surfaces, there are actually a few different definitions: Connolly surface, solvent accessible surface area and van der Waals surface.

A. The solvent-excluded surface (i.e. Connolly surface) is the surface area of the protein as determined by a probe. Typically, the van der Waals radius for each atom is drawn, then the probe is used to (analytically) draw the surface that it can reach. It does not include the radius of the probe.



B. Figure Adapted from: https://link.springer.com/article/10.1007/s00371-017-1397-2
The accessible surface area (ASA) is the area traced off the protein as determined by a probe's center. This includes the radius of the probe itself, so it's different from the connolly surface area.



C. Figure Adapted from: https://en.wikipedia.org/wiki/Accessible_surface_area
The van der Waals surface is the connolly surface plus all the nooks and crannies that the probe cannot reach. This is essentially all the surface area of the atom "spheres",

minus anything that is intersecting with another sphere. The Connolly surface area approaches the van der Waals surface as the probe decreases in size.

There are two general algorithms for SASA calculation. The Shrake-Rupley (S&R) algorithm estimates surface area using points on the surface, while Lee and Richards' (L&R) uses horizontal slices of each atom. See this page for the geometry of L&R (https://freesasa.github.io/doxygen/Geometry.html). From the freeSASA paper (https://f1000research.com/articles/5-189/v1): Both S&R and L&R are pretty straightforward to implement, and both require first determining which atoms are in contact, and then calculating the overlap between each atom and its neighbors. Finding contacts is done using cell lists, which means the contact calculation is an O(N) operation. Both algorithms then treat each atom independently, making also the second part of the calculation O(N). In addition, this second part is trivially parallelizable.

For L&R, instead of slicing the whole protein in one go, each atom is sliced individually. The L&R calculation is thus parameterized by the number of slices per atom, i.e. small atoms have thinner slices than large atoms.

See the MDTraj section 1.IV below for more information about the S&R algorithm.

Pros and cons of each program

I. CPPTRAJ

The `surf` command calculates the surface accessible surface area, while `molsurf` calculates the Connolly surface area. The reference manual indicates `surf` supports decomposed SASA calculations, but it's so approximate that sometimes you will get negative numbers in terms of SASA contributions.

II. VMD

The Visual Molecular Dynamics program (https://www.ks.uiuc.edu/Research/vmd/) is able to calculate SASA and decomposed SASA using the `measure` command (http://www-s.ks.uiuc.edu/Research/vmd/vmd-1.9.1/ug/node136.html). This can be slow to calculate for every frame, but is expected to work fairly well. Explanations of the algorithm are available here and here. In summary, the SASA is calculated by extending the radius of each atom in the target selection by the solvent radius (default = 1.4 Å) and calculating the area of the resultant spheres which is not overlapping another atom.

- VMD `measure` samples N points (where by default N=500) around each atom in the selection, and asks if they are solvent accessible or not. This fraction multiplied by the surface area of the sphere with radius atom radius + probe radius is where the SASA comes from.
- Based on a simple Monte Carlo estimation technique
    - The algorithm samples the surface area of spheres in the atom selection, accepting/rejecting them based on whether or not they fall within the radius of

neighboring atoms. This gives you a "union of spheres" type surface area, which is close but not identical to the area that you get if you use a "rolling ball" type method, say like what MSMS computes.
    - http://www.ks.uiuc.edu/Research/vmd/mailing_list/vmd-l/15227.html
- An example tcl script for SASA calculation in VMD is available here:
    - https://www.ks.uiuc.edu/Research/vmd/mailing_list/vmd-l/att-18670/sasa.tcl
    - Open VMD and source this script from the tk console.
- Likely using S&R

III. MSMS

MSMS (https://ccsb.scripps.edu/msms/) is a widely used program available as a stand-alone program or from VMD and Chimera. MSMS can run SES and SASA calculations for the entire protein only.

- " MSMS consists of four algorithms. The first computes the reduced surface of a molecule from which the second algorithm builds an analytical representation of the solvent-excluded surface that may be self-intersecting. The third algorithm removes all self-intersecting parts. The last algorithm produces a triangulation of the SES."
    - https://onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-0282(199603)38:3%3C305::AID-BIP4%3E3.0.CO;2-Y

IV. MDTraj

From the documentation (https://mdtraj.org/1.9.4/api/generated/mdtraj.shrake_rupley.html) :

- This code implements the Shrake and Rupley algorithm, with the Golden Section Spiral algorithm to generate the sphere points. The basic idea is to great a mesh of points representing the surface of each atom (at a distance of the van der waals radius plus the probe radius from the nuclei), and then count the number of such mesh points that are on the molecular surface – i.e. not within the radius of another atom. Assuming that the points are evenly distributed, the number of points is directly proportional to the accessible surface area (its just 4*pi*r^2 times the fraction of the points that are accessible).
    - Shrake, A; Rupley, JA. (1973) J Mol Biol 79 (2): 351–71.

- There are a number of different ways to generate the points on the sphere – possibly the best way would be to do a little "molecular dynamics" : put the points on the sphere, and then run MD where all the points repel one another and wait for them to get to an energy minimum. But that sounds expensive.

- This code uses the golden section spiral algorithm (picture at http://xsisupport.com/2012/02/25/evenly-distributing-points-on-a-sphere-with-the-golden-sectionspiral/) where you make this spiral that traces out the unit sphere and then put points down equidistant along the spiral. It's cheap, but not perfect.

- The gromacs utility g_sas uses a slightly different algorithm for generating points on the sphere, which is based on an icosahedral tessellation. roughly, the icosahedral

tesselation works something like this
http://www.ziyan.info/2008/11/sphere-tessellation-using-icosahedron.html

V. NACCESS

Naccess (http://www.bioinf.manchester.ac.uk/naccess/) may work well for decomposed SASA but the process to obtain the program is cumbersome. In order to access this program, you have to first snail-mail a signed confidentiality agreement, then send a separate follow-up email to simon.hubbard@manchester.ac.uk to get a decryption key.

- Uses L&R

VI. freeSASA

The freeSASA python program (https://freesasa.github.io/) may work well and has the ability to run decomposed SASA calculations. Can be installed via pypi or from source.
From the documentation:

- FreeSASA is a command line tool, C-library and Python module for calculating solvent accessible surface areas (SASA). Casual users can calculate SASA directly from a PDB file with no configuration overhead using sensible default parameters. Advanced users can configure all calculation parameters and also use the C and Python APIs to perform calculations directly on arrays of coordinates.
- Features PyMol atom selection
- Can choose L&R or S&R

2. Results from SASA calculation tests using various programs:

I. Overall or Total SASA Calculations

Table 2. Total SASA values using a solvent probe radius of 1.4 Å for the Wild-Type BdpA protein.

SASA Program	**Unfolded Total SASA (Å^2)**	**Folded Total SASA (Å^2)**
CPPTRAJ-LCPO	5467	4820
MDTraj	5340	4540

II. Decomposed SASA Calculations

Table 1. Decomposed SASA values using a solvent probe radius of 1.4 Å for the closed SARS-COV2 spike protein glycans corresponding to N234. VMD and MDTraj results are comparable and appear reasonable, while the CPPTRAJ data is incorrect.

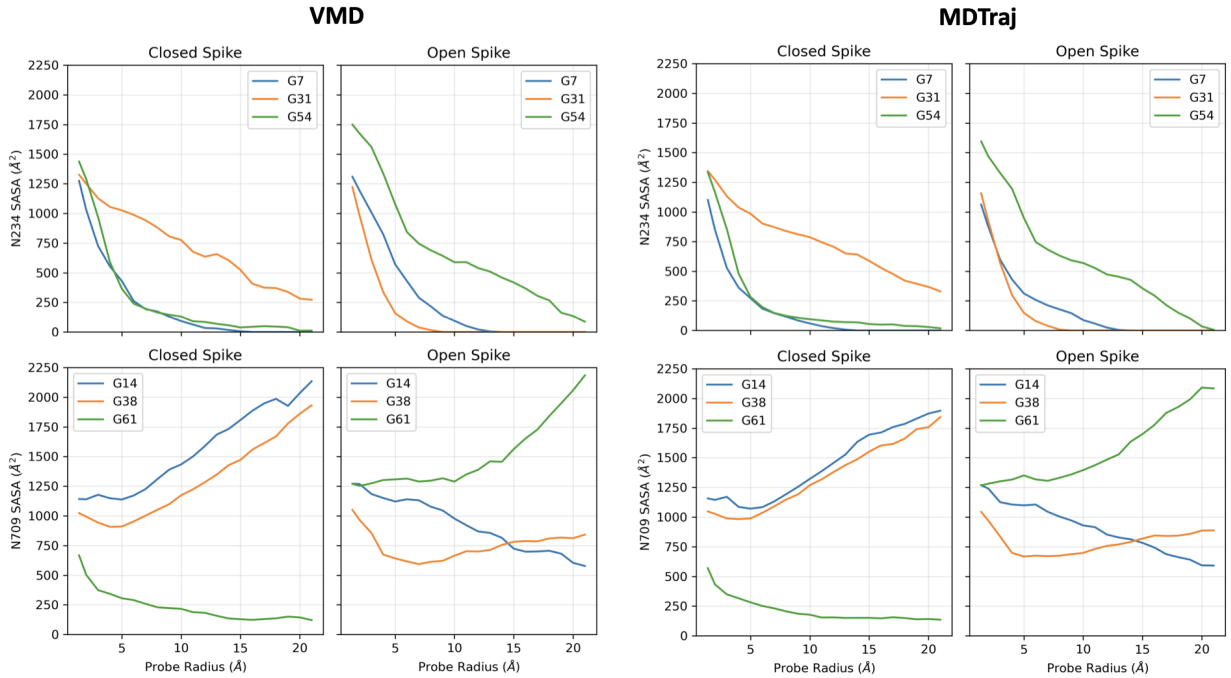
SASA Program	**G7 (Å^2)**	**G31 (Å^2)**	**G54 (Å^2)**

CPPTRAJ-LCPO	-96.23	-74.49	-82.25
VMD	1273.33	1326.08	1437.57
MDTraj	1101.20	1343.31	1336.05

Table 2. Decomposed SASA values using a solvent probe radius of 1.4 Å for the Wild-Type BdpA Protein. Note that the differences between folded and unfolded state remain relatively constant, suggesting that differences between values are somewhat conserved.

SASA Program	**Unfolded HC SASA (Å^2)**	**Folded HC SASA (Å^2)**	**Unfolded Backbone Amide SASA (Å^2)**	**Folded Backbone Amide SASA (Å^2)**
CPPTRAJ-LCPO	2248	1750	1020	825
MDTraj	3630	3070	620	400

III. Decomposed SASA as a Function of Probe Radius



When accessing decomposed SASA of individual glycans on the SARS-CoV-2 trimeric spike protein as a function of probe radius, the results are fairly consistent between VMD and MDTraj. What both also happen to show is that with N234, which is critical for RBD conformational changes, the SASA decreases with increasing probe radius, which can be explained by the larger probe being able to access less nooks and crannies on the surface of

the glycan when the probe radius increases. For N709, which is expected to be more exposed, the SASA initially decreases, but then some N709 glycans of the trimer actually increase in SASA.

This may be due to a few reasons. If considering a sphere as your molecule, an approximation of the calculation of SASA can be thought of as a probe sphere of a specified radius is used to "roll" over your molecule and the calculated surface is determined from the center of this rolling sphere. With this logic, a larger probe radius will lead to a larger calculated area for a sphere; however, this is not true in general, such as with a cylinder. In these cases, the SASA will depend on the radius of the probe and the diameter of the cylinder. Note that also, for a probe radius near infinity, you'll get the area of a convex hull (https://en.wikipedia.org/wiki/Convex_hull) surrounding the structure. For a probe radius of zero, you'll get the surface area corresponding to the "union of balls". A probe radius of >4.25 Å is rapidly heading towards the convex hull case (https://www.ks.uiuc.edu/Research/vmd/mailing_list/vmd-l/15252.html).

3. Our final conclusions and overall recommendations:

Overall, we found that for overall SASA calculations, any of the above mentioned programs should work just fine. When considering partial or decomposed SASA calculations, using MDTraj, VMD, or freeSASA should all work well and have comparable results. The only outlier is the CPPTRAJ LCPO method, which can give negative and highly inaccurate decomposed SASA results. The CPPTRAJ LCPO method was not parameterized for decomposed SASA but is a reasonable approximation for total SASA calculations, which are comparable to, but faster than the CPPTRAJ Connolly surface method.