

# Exact analytical algorithm for solvent accessible surface area and derivatives in implicit solvent molecular simulations on GPUs

Xin Cao,<sup>†,‡</sup> Michelle H. Hummel,<sup>¶</sup> Yuzhang Wang,<sup>§</sup> Carlos Simmerling,<sup>§,‡</sup> and Evangelos A. Coutsias<sup>\*,†,‡</sup>

<sup>†</sup>*Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, United States*

<sup>‡</sup>*Laufer Center for Physical and Quantitative Biology, Stony Brook University, Stony Brook, NY 11794, United States*

<sup>¶</sup>*Sandia National Laboratories, Albuquerque, NM 87123, United States*

<sup>§</sup>*Department of Chemistry, Stony Brook University, Stony Brook, NY 11794, United States*

E-mail: [evangelos.coutsias@stonybrook.edu](mailto:evangelos.coutsias@stonybrook.edu)

## Abstract

In this paper, we present dSASA (differentiable SASA), an exact geometric method to calculate solvent accessible surface area (SASA) analytically along with atomic derivatives on GPUs. The atoms in a molecule are first assigned to tetrahedra in groups of four atoms by Delaunay tetrahedrization adapted for efficient GPU implementation and the SASA values for atoms and molecules are calculated based on the tetrahedrization information and inclusion-exclusion method. The SASA values from the numerical icosahedral-based method can be reproduced with more than 98% accuracy for both proteins and RNAs. Having been implemented on GPUs and incorporated

into the software Amber, we can apply dSASA to implicit solvent molecular dynamics simulations with inclusion of this nonpolar term. The current GPU version of GB/SA simulations has been accelerated up to nearly 20-fold compared to the CPU version and it outperforms LCPO as the system size increases. The performance and importance of the nonpolar part in implicit solvent modeling are demonstrated in GB/SA simulations of proteins and accurate SASA calculation of nucleic acids.

## Introduction

An accurate description of the solvent environment is essential for biomolecular modeling, as biological machines are functioning in an aqueous environment. The solute-solvent interactions and the rearrangement of water molecules induce the change of molecular shapes and the solvation free energy  $\Delta G_{sol}$ .<sup>1</sup> Including water molecules in explicit solvent simulations produces more accurate simulation results at the cost of redundant calculation for the pairwise interactions between water molecules, while the friction induced by collisions with water molecules can further slow down the conformational sampling of the solute. In implicit solvent models, the solvent is treated as a continuum and the system is simulated without explicit water molecules. The major advantage of implicit solvent models is the ability to explore the conformational space more efficiently, which can find applications in protein folding studies and structure prediction,<sup>2,3</sup> solvation free energy calculation using Poisson-Boltzmann surface area (PBSA)<sup>4</sup> or generalized Born surface area (GBSA),<sup>5</sup> and binding free energy calculation using molecular mechanics Poisson-Boltzmann surface area (MM-PBSA).<sup>6</sup>

In implicit solvent modeling, the solvation free energy can be decomposed into polar and nonpolar contributions<sup>7</sup> and the expression is

$$\Delta G_{sol} = \Delta G_{pol} + \Delta G_{cav} + \Delta G_{vdW} = \Delta G_{pol} + \Delta G_{np},$$

where the polar term  $\Delta G_{pol}$  gives the difference between the work of uncharging the solute in vacuum and the work of charging the solute in solvent,  $\Delta G_{cav}$  denotes the accommodation of cavity in the solvent for solute and  $\Delta G_{vdW}$  represents the van der Waals interactions between solute and solvent. The Poisson-Boltzmann (PB) equation models the electric field for the electrostatic interactions between solvent and solute<sup>8</sup> and can be used to compute  $\Delta G_{pol}$ . The  $\Delta G_{cav}$  and  $\Delta G_{vdW}$  are usually combined into one nonpolar term  $\Delta G_{np}$ . It is considered to be proportional to the number of atoms in the solute having direct contact with solvent molecules and can be estimated in terms of the solvent accessible surface area (SASA).<sup>9</sup> Although it was pointed out that incorporation of the volume term into the computation can provide a more complete description of the nonpolar part,<sup>10,11</sup> the surface area based methods can still have good performance in the prediction of the native-like conformations of proteins and the estimation of ligand-binding affinities.<sup>12</sup>

In implicit solvent molecular dynamics (MD) simulations, most of the effort was devoted to the development of the dominant polar part. Due to the complexity of solving PB equation, the generalized Born (GB) method<sup>5</sup> was proposed to approximate PB equation with simple functions. The improvement of GB model on proteins<sup>13-15</sup> and the variant to nucleic acids<sup>16</sup> and its implementation on graphics processing units (GPUs)<sup>17</sup> helped GB achieve popularity in recent years. The main goal of GB model is to reproduce the “perfect” effective radii and the free energies estimated from PB model. In particular, the pairwise decomposition version of GB<sup>18</sup> is ideal for implementation on GPU with parallel computing. Although the performance of GB is improving, poor folding stabilities were observed in folding studies,<sup>2</sup> structure predictions<sup>3</sup> and de novo designed peptides with high helicity.<sup>19</sup> As hypothesized in,<sup>2</sup> the ignored nonpolar solvation term might result in instability; this was supported by improved fold stability when an approximate SASA-based nonpolar term was added to the same GB model.<sup>20</sup> As shown in that paper, the inclusion of the nonpolar term in the GB/SA MD simulations can produce more stable trajectories and better simulated melting temperatures than GB-only simulations.

There exist several methods to calculate the molecular surface area on CPUs and GPUs. However, only a few of them can be applied to MD simulations because most of them lack reliable derivatives for atoms. The solvent accessible surface was proposed by Lee and Richards in.<sup>21</sup> A numerical implementation by Connolly in<sup>22</sup> is computationally expensive and lacks analytical derivatives. The derivatives are unreliable or unavailable in other methods, such as grid based, neighbor-counting or machine learning.<sup>23</sup> The approximation methods on GPU have similar issues.<sup>24,25</sup> In CHARMM software, GBMV2/SA<sup>26</sup> can reproduce the molecular surface and has been implemented on GPUs.<sup>27</sup> However, the computation is based on grid points so lower resolution of the grids may generate less reliable results and higher resolution may result in slower simulation speed. The first available method for approximating surface area along with derivatives in Amber is LCPO<sup>28</sup> which is able to estimate SASA based on the neighbors of atoms. As the MD simulations in Amber are now primarily on GPUs,<sup>17,29</sup> the data transfer between CPU and GPU deteriorates the performance because LCPO is implemented only on CPUs. Recently, another approximate pairwise method with a faster speed (up to 30 times faster), called pwSASA, was proposed and it achieved a comparable accuracy to LCPO according to the testing of proteins in.<sup>20</sup> The main purpose of pwSASA was to investigate the impact of the nonpolar term on the MD simulations of proteins with a simple and fast pairwise approximation suitable for running entirely on GPUs. The simple form of the functions provided faster simulation speed and did prove the significance of nonpolar term in the simulations. However, the computed SASA values had various correlation coefficients from 0.6 to 0.9 compared with the numerical computation for a number of proteins, which is partly because the higher order interactions were ignored. Besides the varied performance for different protein systems, a parametrization for nucleic acids or small molecule ligands is not available in pwSASA because it was trained exclusively based on proteins.

As every atom in a molecule can be represented by a sphere, the surface areas have also been studied by computational geometry methods. One reliable and accurate method was

developed based on the Alpha Complex construction<sup>30</sup> by which the geometric descriptors of molecules can be computed through union of balls. The Alpha Complex is obtained with the atomic coordinates and radii information and can be applied to decompose the space into small segments through which the surface areas are computed by using Gauss-Bonnet theorem<sup>31</sup> or inclusion-exclusion formulas.<sup>32-34</sup> Recently, the extension to other geometric measures, such as volumes, mean and Gaussian curvatures, was proposed in<sup>35</sup> where the parallel implementation can achieve speed improvement over the serial version. However, the full implementation of such method on GPUs is still unavailable because each step in the method requires substantial effort to utilize the parallel computing property to improve performance.

Here we present dSASA, a new method to calculate SASA and its derivatives with respect to atomic coordinates using Alpha Complex theory and inclusion-exclusion method. This fully analytical and accurate algorithm has been implemented on GPUs and incorporated in Amber software. The basic background and the procedures in the method will be first shown. The results for the assessment of the method are next given: the accuracy of estimation is examined by several protein systems with diverse topologies, the speed of the GPU version in GB/SA simulations is compared with the CPU version and other methods, the performance in GB/SA simulations on proteins by including this nonpolar term and the validation on RNAs. The findings prove the valuable role of an accurate nonpolar term in improving the ability to predict native-like structures using the implicit solvent simulations.

## Methods

### Theory and estimations for nonpolar solvation

In the implicit solvent model, the nonpolar solvation term is usually SASA-based.<sup>5</sup> The free energy is proposed to be proportional to SASA with a surface tension parameter ( $\gamma$ ):  $\Delta G_{np} = \gamma SASA$ . The surface tension is assumed to be identical for all atoms.

The relatively accurate SASA values can be calculated by ICOSA numerical method<sup>36</sup> ( $gbsa = 2$  in Amber), in which starting from an icosahedron a water probe with radius 1.4 Å is recursively rolled on the van der Waals surface of the molecule. The implementation of this method in current Amber MD simulations is unavailable because of the absence of atomic forces. The linear combinations of pairwise overlaps (LCPO;  $gbsa = 1$  in Amber) is the first stable algorithm used in GB/SA MD simulations in Amber. The neighbor list of a central atom is used to subtract the pairwise overlapping from its isolated sphere areas, which results in the computation complexity  $O(N^2)$ , so it will slow down quadratically as the size of the molecule increases. In the GPU version of GB/SA MD with this method, the data transfer between CPU and GPU decelerates the simulations as its implementation is CPU/GPU hybrid. The pairwise approximation of SASA (pwSASA;  $gbsa = 3$  in Amber)<sup>20</sup> is the recently developed algorithm to calculate SASA that is suitable for implementing on GPUs. The idea is similar to LCPO but only considers the interactions between two atoms, which would achieve a faster speed but result in inaccuracy especially when the molecule contains some unusual atoms. The parameters for the atom types are trained exclusively on proteins so the application to other systems, such as nucleic acids or small molecules, is limited.

## **dSASA: an analytical method for SASA**

Here we introduce *dSASA*, a geometric method ( $gbsa = 4$  in Amber) to compute SASA. As has been explored in,<sup>28,37</sup> to be consistent with other methods, we assume that the heavy atoms in molecules can have a good approximation to the molecular SASA, reducing computational cost substantially. The atomic radii for four common elements (C, 1.7 Å; O, 1.5 Å; N, 1.55 Å; S, 1.8 Å) are used in ICOSA, pwSASA and dSASA, while LCPO is using different radii (C, 1.7 Å; O, 1.6 Å; N, 1.65 Å; S, 1.9 Å). Every atom can be represented by a weighted point  $p'_i = (p_i, d_i)$  in the point set  $\mathbb{A} \subset \mathbb{R}^3 \times \mathbb{R}$ , which contains the 3D coordinates  $p_i$  in space and one point weight  $d_i$ . The radius of water probe is set to 1.4 Å, and the weight

$d_i = (\text{atom\_radius}_i + 1.4)^2$ . Here we present the key steps, while a complete description of the analytical development of the method is given in.<sup>33,34</sup> The method has three main steps and the depiction of the method is shown in Figure 1.

**Step 1.** Calculate the 3D weighted Delaunay Tetrahedrization (wDT) of all points through which the points are assigned into tetrahedra. This step has computation complexity  $O(N \log N)$  and is the dominating part of the method.

**Step 2.** Create and classify the dual complex  $\mathcal{C}$  which denotes the possible interactions among close atoms. The complex  $\mathcal{C}$  contains the lists of simplices, such as the vertices (the atoms), edges (interactions between two atoms), triangles (interactions among three atoms) and tetrahedra (interactions among four atoms) from the tetrahedrization. These simplices are then filtered and classified into interior and exterior (denoted by  $\partial\mathcal{C}$ ) based on their connection information.

**Step 3.** Compute the surface areas using Laguerre intersection cells and inclusion-exclusion method based on the exterior simplices  $\partial\mathcal{C}$ .

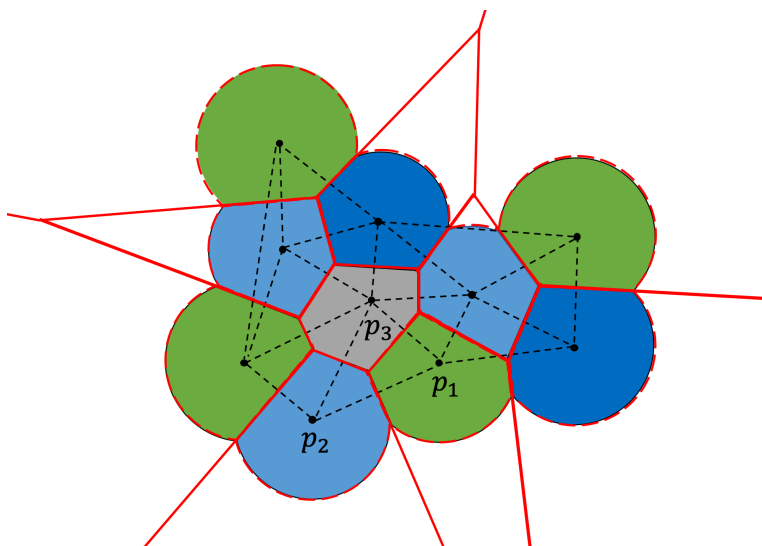


Figure 1: Depiction of the method in 2D. Every solid circle is the expanded atom including the water probe, black points show the center of atoms. The dashed black lines denote part of the tetrahedrization of the molecule. The atoms in green and blue (such as  $p_1, p_2$ ) are exterior, grey atom  $p_3$  is interior. The boundary of Laguerre cells of atoms are shown in solid red lines, and exterior boundary of Laguerre cells of molecule in solvent is represented by the dashed red curves which correspond to SASA.

In the following, we show the process to compute the atomic SASA based on the exterior simplices  $\partial\mathcal{C}$  because only the atoms having direct contacts with solvent contribute to the surface areas. The simplices have been classified into interior and exterior at Step 2. The Laguerre diagram of the point set is conjugate to the wDT. The space can be decomposed into cells by the conjugated Laguerre diagram as shown in Figure 1. The Laguerre cell for a weighted point (atom)  $p'_i$  is defined by

$$L_i = \{x \in \mathbb{R}^3 : |p_i - x|^2 - d_i \leq |p_j - x|^2 - d_j, \forall p'_j \in \mathbb{A}\}.$$

Every weighted point can also be treated as a ball  $B_i = \{x \in \mathbb{R}^3 : |p_i - x|^2 - d_i \leq 0\}$ , and the union of these balls  $B = \cup B_i$  forms the space filling model of  $\mathbb{A}$ . As shown in Figure 1, the Laguerre cells of exterior atoms can be unbounded or larger than the real size of the atoms, then a more realistic representation for an atom is given by the Laguerre-intersection cell, the intersection part of the Laguerre cell and the ball:  $LI_i = L_i \cap B_i$ . The boundary area of

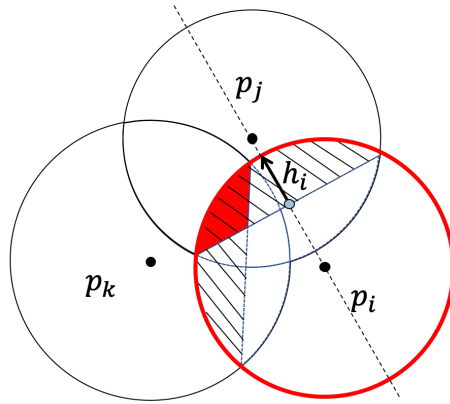


Figure 2: Inclusion-exclusion process of  $p_i$ . The red circle denotes the standalone ball  $S_i$  of  $p_i$ ; the shaped region indicates the intersection  $S_{ij}$  and  $S_{ik}$  with  $p_j, p_k$ ; the small red region is the intersection of three atoms  $S_{ijk}$ .

the Laguerre intersection part  $LI_i$  of an exterior atom is the SASA value  $\mathcal{S}^i = surf(LI_i)$ .

Suppose there are  $k$  points in a subset  $T \subset \mathbb{A}$ . The number of points in  $T$  is  $|T| = k$  and the centers of points are denoted by  $T'$ . The convex hull (*conv*) of points in  $T'$  is written as



$\sigma_T$ , and the term  $S_T$  is the surface area of the intersection of the balls in  $T$ . By applying the inclusion-exclusion method shown in Figure 2, the SASA of the molecule is

$$\mathcal{S} = \sum_{\sigma_T \in \partial \mathcal{C}} (-1)^{k+1} c_T S_T, \quad k = 1, 2, 3.$$

The coefficients  $c_T$  are given by the corresponding simplices. When  $|T| = 1$ , i.e. a vertex  $v_i$ :  $c_T = \Omega_T$  is the fraction of the ball  $i$  outside the tetrahedra in  $\mathcal{C}$ .  $\Omega_T$  is the normalized outer solid angle subtended by the union of tetrahedra in  $\mathcal{C}$  containing  $v_i$ . When  $|T| = 2$ , i.e. an edge  $e_{ij}$ :  $c_T = \Phi_T$  is normalized outer dihedral angle of the union of tetrahedra in  $\mathcal{C}$  which contain the edge  $e_{ij}$ . When  $|T| = 3$ , i.e. a triangle  $t_{ijk}$ :  $c_T = 1$  or  $0.5$  is the fraction of  $S_T$  that is outside the union of tetrahedra in  $\mathcal{C}$ . Here  $v_i = p_i$ ,  $e_{ij} = \text{conv}(\{p_i, p_j\})$ , and  $t_{ijk} = \text{conv}(\{p_i, p_j, p_k\})$ .

The contribution of individual atom  $p'_i$  to the molecular surface area is

$$\mathcal{S}^i = \Omega_i S_i^{(i)} - \sum_{e_{ij} \in \partial \mathcal{C}} \Phi_{ij} S_{ij}^{(i)} + \sum_{t_{ijk} \in \partial \mathcal{C}} c_{ijk} S_{ijk}^{(i)}, \quad (1)$$

where  $S_T^{(i)}$  is the contribution of  $S_T$  to  $\mathcal{S}^i$ , and  $\sum_{i=1}^n \mathcal{S}^i = \mathcal{S}$ .

Then the corresponding derivative for the points can be obtained on the basis of the atomic surface area with respect to the atomic coordinates in the molecule as

$$\nabla \mathcal{S}^i = \nabla \Omega_i S_i^{(i)} - \sum_{e_{ij} \in \partial \mathcal{C}} (\nabla \Phi_{ij} S_{ij}^{(i)} + \Phi_{ij} \nabla S_{ij}^{(i)}) + \sum_{t_{ijk} \in \partial \mathcal{C}} c_{ijk} \nabla S_{ijk}^{(i)}. \quad (2)$$

The terms in the equations are calculated as follows. When  $|T| = 1$ ,  $S_T^{(i)} = 4\pi d_i$ . Let  $\mathcal{I}_1$  be the set of tetrahedra in  $\mathcal{C}$  to which the point  $p'_i$  belongs to. For  $\zeta_I \in \mathcal{I}_1$  define  $\omega^I$  as the normalized inner solid angle subtended by the tetrahedron  $\zeta_I$  from the point  $p'_i$ . Then

$$\Omega_T = 1 - \sum_{\zeta_I \in \mathcal{I}_1} \omega^I, \quad \text{with the derivative } \nabla \Omega_T = - \sum_{\zeta_I \in \mathcal{I}_1} \nabla \omega^I. \quad (3)$$

The normalized inner solid angle,  $\omega^I$ , of a tetrahedron  $p_i p_j p_k p_l$  subtended by the vectors  $\mathbf{a} = p_j - p_i$ ,  $\mathbf{b} = p_k - p_i$ , and  $\mathbf{c} = p_l - p_i$  is shown in the left of Figure 3 and given by the equation

$$\omega^I = \frac{1}{2\pi} \arctan \left( \frac{|\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})|}{abc + (\mathbf{a} \cdot \mathbf{b})c + (\mathbf{a} \cdot \mathbf{c})b + (\mathbf{b} \cdot \mathbf{c})a} \right),$$

where  $a = |\mathbf{a}|$  is the length of  $\mathbf{a}$  and likewise for  $b$  and  $c$ .

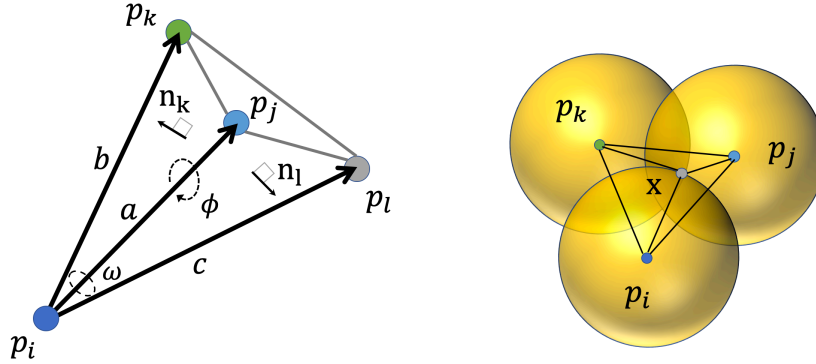


Figure 3: Left: Tetrahedron formed by points  $p_i, p_j, p_k, p_l$ . Right: Center  $\mathbf{x}$  of the intersection of three balls and the resulting tetrahedron.

When  $|T| = 2$ , the intersection of two atoms  $p'_i, p'_j$  is shown in Figure 2.  $S_T^{(i)} = 2\pi\sqrt{d_i}h_i$ . Let  $\mathcal{I}_2$  be the set of tetrahedra in  $\mathcal{C}$  containing the edge  $\sigma_T$ . For  $\zeta_I \in \mathcal{I}_2$  define  $\phi^I$  as the normalized inner dihedral angle of  $\zeta_I$  along  $\sigma_T$ . Then

$$\Phi_T = 1 - \sum_{\zeta_I \in \mathcal{I}_2} \phi^I.$$

The normalized inner dihedral angle between planes  $p_i p_j p_k$  and  $p_i p_j p_l$  with normals  $\mathbf{n}_k$  and  $\mathbf{n}_l$  is shown in the left of Figure 3 and given by

$$\phi^I = \frac{\arccos(\mathbf{n}_k \cdot \mathbf{n}_l)}{2\pi}, \text{ with } \nabla \phi^I = \frac{-\nabla(\mathbf{n}_k \cdot \mathbf{n}_l)}{2\pi\sqrt{1 - (\mathbf{n}_k \cdot \mathbf{n}_l)^2}}. \quad (4)$$

Then the derivative terms for Eq.(2) are

$$\nabla S_T^{(i)} = 2\pi\sqrt{d_i}\nabla h_i, \quad \nabla\Phi_T = -\sum_{\zeta_I \in \mathcal{I}_2} \nabla\phi_T^I. \quad (5)$$

When  $|T| = 3$ , consider  $T = \{p'_i, p'_j, p'_k\}$ . The surface area of the common intersection of three balls can be written as a weighted sum of the surface area of the single and double intersections. If  $p'_i, p'_j, p'_k$  have a non-empty intersection then there are two points in common with the surfaces of all three balls, shown in the right of Figure 3. Denote one of the two points  $\mathbf{x}$ , define  $p'_x = (\mathbf{x}, 0) \in \mathbb{R}^3 \times \mathbb{R}$ , and let  $T_x = \{p'_i, p'_j, p'_k, p'_x\}$ . Let  $S_2^x$  be the set of edges defined by  $\sigma_{T_x}$  and  $S_1^x$  the set of vertices in  $\sigma_{T_x}$ . The atomic surface area of  $p'_i$  from the intersection of  $p'_i, p'_j, p'_k$  is given by

$$\frac{1}{2}\mathcal{S}_T^{(i)} = \Phi_{ij}^x S_{ij}^{(i)x} + \Phi_{ik}^x S_{ik}^{(i)x} - \Omega_i^x S_i^x, \quad (6)$$

where  $\Phi_{ij}^x$  is the normalized dihedral angle of  $\sigma_{T_x}$  along the edge  $\sigma_{ij}$ ,  $\Omega_i^x$  is the normalized solid angle of  $\sigma_{T_x}$  subtended from  $p_i$  and similarly for other combinations  $i, j, k$ . The derivative of this term is

$$\frac{1}{2}\nabla\mathcal{S}_T^{(i)} = \nabla\Phi_{ij}^x S_{ij}^{(i)x} + \Phi_{ij}^x \nabla S_{ij}^{(i)x} + \nabla\Phi_{ik}^x S_{ik}^{(i)x} + \Phi_{ik}^x \nabla S_{ik}^{(i)x} - \nabla\Omega_i^x S_i^x. \quad (7)$$

By substituting Eqs. (3), (4), (5), (6) and (7) into Eqs. (1) and (2), we can obtain the surface area and the derivative for every atom. More details of the description and the equations can be found in<sup>33,34</sup>. However, the atomic derivatives may become discontinuous when atoms are approaching. Through detailed analysis in<sup>33</sup>, it can be seen that most common type of discontinuities occur when two atoms become externally tangent. We examine here the simplest case where only two atoms with extended radii  $r_i = 2.9, r_j = 3.1 \text{ \AA}$  are considered to show the possible singularity in the calculation. The changing process of the SASA values and the atomic derivatives is given in Figure 4. In the left, the summation

of SASA values is continuous as two atoms are approaching from a long distance and the minimal value occurs when two atoms are internally tangent. In the right, the magnitude of the atomic derivatives has a leap when two atoms are externally tangent at which the singularity occurs. As investigated in<sup>33</sup>, the singularity rarely happens when the solvent radius is 1.4 Å. In the implementation, we need one threshold value  $\epsilon$  to determine the external tangency if  $||p_i - p_j| - (r_i + r_j)| < \epsilon$ . When  $\epsilon = 10^{-6}$ , the external tangency was not detected after simulating a few systems for thousands of steps. When  $\epsilon = 10^{-3}$ , it may have one such case after dozens of steps simulation. On the other hand, the magnitude of the leap in the atomic derivatives is less than 20 Å<sup>2</sup> and the surface tension is usually less than 0.01 kcal/(mol·Å<sup>2</sup>), so the impact of such singularity is insignificant compared to the overall forces for exterior atoms whose magnitude can be more than 10 kcal/(mol·Å<sup>2</sup>). Given the little chance of happening and the tiny impact of the singularity on the overall forces of the atoms, we can expect that the discontinuity in SASA calculation will have negligible impact on the trajectories in GB/SA simulations.

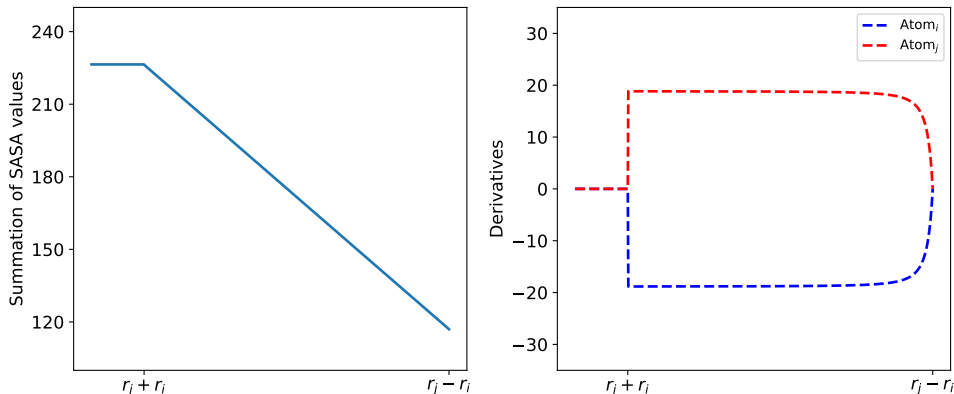


Figure 4: Trend of SASA and the magnitude of atomic derivatives as the distance of two atomic centers is changing. The x-axis shows the distance between the atomic centers. The summation of SASA values is on the left, while the atomic derivatives are on the right.

## Implementation of dSASA on GPU

As the implementation of MD in Amber has been done on GPUs<sup>17,29</sup>, our main concern here is to implement the method on GPUs to speed up the calculation, making it applicable to longer time MD simulations, so we discuss more details about the necessary treatments on GPUs below.

### Weighted Delaunay Tetrahedrization

In the wDT, all possible connections among local points can be detected. Based on the coordinates and weights, four local points are assigned to a tetrahedron having the property that no other points will be inside the circumsphere of this tetrahedron. The wDT can be sequentially computed on CPUs with complexity  $O(N \log N)$ . One parallel algorithm on GPUs to generate exact wDT for a large dataset remains a challenge. The typical GPU implementation of wDT often obtains a near-Delaunay tetrahedrization followed by transformation to CPU to generate the valid wDT. However, an algorithm gReg3D<sup>38</sup> which is able to compute the exact wDT on GPUs was implemented a few years ago. In the algorithm, all points are first included into a cube with an appropriate size and next the points will be assigned to smaller cubes. The local tetrahedrization for these small cubes are calculated in parallel, followed by checking the consistency and applying necessary modifications to achieve the final wDT. The largest size of the initial cube in this algorithm is 512, so the point set with a greater size will be rescaled to fit into the cube, which may introduce a few errors. As the SARS-CoV-2 spike protein containing nearly 4000 residues typically occupies a cube with size 256, this algorithm can compute an exact wDT for proteins with thousands of residues on GPUs. As shown in Figure 1, the black points are the center of atoms and dashed lines connecting black points denote the edges in the tetrahedrization. The purpose of this step is to obtain the dual complex  $\mathcal{C}$  and the conjugated Laguerre diagram. In this diagram, every point is represented by a cell enclosed by the red solid lines and dashed curves, then the surface area can be calculated with such information.

## Extraction of exterior simplices and calculation of surface areas

The information of close atoms in the tetrahedra has been achieved in the wDT above. However, some unrealistic connections and the interior contacts will not have contribution to surface areas. Then we need to extract the exterior simplices  $\partial\mathcal{C}$  containing the feasible vertices, edges and triangles along with the connections among them, which will finally be used to calculate SASA. The extraction of simplices on CPUs is serial and cannot be implemented on GPUs directly. Here we provide more details for the extraction of simplices on GPUs.

We first need to create lists of unique triangles, edges and vertices with connection information. Take one tetrahedron as an example. It contains 4 triangles as faces, 6 edges and 4 vertices. One element in the list of triangles contains the information of three vertices and the tetrahedra it belongs to, one edge element contains the information of two vertices and the triangles and tetrahedra it is in, while one vertex element has the information of the associated edges and tetrahedra.

The exported information from gReg3D provides the vertex indices of all tetrahedra, so we can pull out all the information of edges and triangles in tetrahedra by taking advantage of the parallelism on GPUs. One list of all possible triangles is created and every element has three vertices and the associated tetrahedra. The list of all possible edges is constructed with each element containing two vertices and the associated tetrahedra. Suppose there are  $N$  tetrahedra, then  $4N$  possible triangle elements and  $6N$  possible edge elements are in the lists respectively. However, some duplicate elements need to be filtered out to create the lists of unique elements. Take the list of triangles as an example shown in Figure 5. We first do lexical sorting for the list of triangles by the first three indices, bringing the same triangles next to each other. Next, with parallel computing, we can identify the number of the associated tetrahedra for a triangle. One triangle can belong to at most two tetrahedra. In the following assignment, every element will check the three vertices of the next element in the list, if three vertices are identical, then we will assign '1' to the newly created array

“idxTri” which indicates the triangle is duplicate, otherwise we assign ‘0’. We then can identify the unique triangle elements in the list and record the indices in one array “Tri”: a ‘1’ in “idxTri” is a new element, and the second ‘0’ in two consecutive ‘0’s indicates a new element as well. Now we have the indices for all unique triangles and the number of associated tetrahedra, then we can create the data structure “TriList” for these triangles. When creating triangles, we extract a list of edges and assign the associated triangle indices to this new list, which will provide partial information to edges. Following the similar process above, we can create the lists of unique edges and unique vertices respectively.

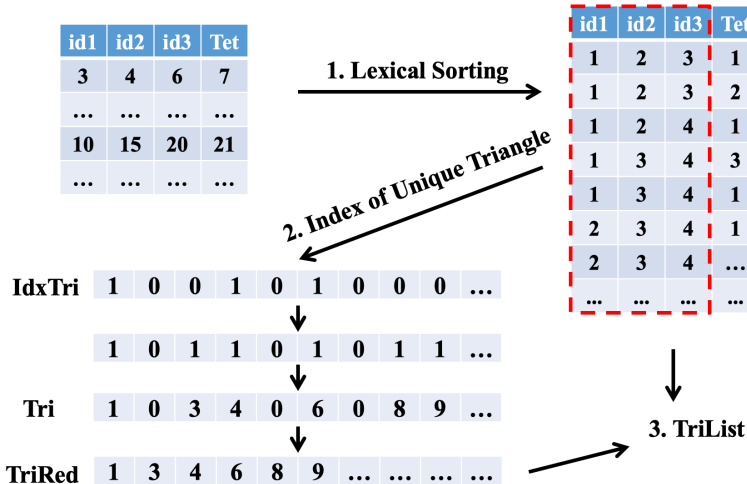


Figure 5: Procedures to create the list for unique triangles.

With the lists for vertices, edges and triangles, we next can classify them into interior and exterior, and record the indices of exterior simplices. The basic rule is: the triangle with at most one associated tetrahedron is exterior, and one edge is exterior only if its two vertices are both exterior. As shown in Figure 1, the atoms in green and blue are exterior while the grey atom  $p_3$  is interior whose SASA value will be 0, the edge  $p_1p_2$  is exterior and the edges  $p_1p_3, p_2p_3$  are interior. For a molecule with 1700 atoms, the numbers of tetrahedra, triangles and edges are more than 11200, 22600 and 13000 respectively. These simplices can be independently implemented in parallel on a GPU to speed up the classification. Given the lists of exterior vertices, edges and triangles, we then can calculate atomic surface areas

and gradients by Eqs. (1) and (2) in parallel and assign the values to individual atoms.

## Simulated protein systems, RNA systems

Trp-cage variant Tc5b (PDB code, 1L2Y<sup>39</sup>) has 20 residues with the sequence NLYIQWLKD GGPSSGRPPPS, and the burial of the hydrophobic tryptophan side chain provides the driving force for its folding. It contains an  $\alpha$ -helix, a  $3_{10}$ -helix and a C-terminal PP<sub>II</sub> helix and the Trp indole ring encapsulated in a cluster of Pro rings. It was designed and solved via NMR experiment.

Homeodomain variant (PDB code, 2P6J<sup>40</sup>) contains 52 residues with sequence MKQW SENVEEKLKEFVKRH <sup>$\delta$</sup> QRITQEELH <sup>$\delta$</sup> QYAQRLGLNEEAIRQFFEEFEQRK. It is a variant of *Drosophila melanogaster* engrailed homeodomain and was solved by NMR. It consists of three  $\alpha$ -helices connected by loop regions.

The 14-mer cUUCGg tetraloop hairpin RNA (PDB code, 2KOC<sup>41</sup>) is NMR-solved model and contains 14 bases with sequence GGCACUUCGGUGCC, with the common and highly stable UUCG loop.

The stem loop C 5'AUA3' triloop of Brome Mosaic virus RNA (PDB code, 1ESH<sup>42</sup>) is composed of 13 bases with sequence GGUGCAUAGCACC. It is designed to contain the triloop AUA in the middle and is an NMR-solved model.

The CD experiments provide the melting curves for Trp-cage<sup>39</sup> and the melting temperature for homeodomain variant.<sup>40</sup> These two protein systems were studied in<sup>20</sup> and the ab initio folding experiments in<sup>2</sup> using the same force field and solvent model, providing a good reference to quantify the possible improvement by addition of a nonpolar solvation term.

## Details of MD simulation

In the GB and GB/SA MD simulations, replica exchange molecular dynamics (REMD) was applied to enhance the efficiency of sampling. The setting for the system is as follows: all bonds involving hydrogen were added SHAKE constraints; Langevin dynamics ( $ntt = 3$ )



with  $1 \text{ ps}^{-1}$  collision frequency was used; the time step was 4 fs by following the protocol in<sup>43</sup> through which the masses of hydrogen atoms can be repartitioned; exchanges between adjacent temperature replicas were attempted every 1 ps; conformations were extracted every 0.1 ns. Trp-cage and homeodomain were parametrized by ff14SBonlysc<sup>44</sup> with GBNeck2<sup>15</sup> and mbondi3 radii. The surface tension parameter was extensively tested in pwSASA<sup>20</sup> for proteins and 5, 7 and 10 cal/(mol·Å<sup>2</sup>) were working properly in Amber. Then we implemented the GB/SA simulations for LCPO, pwSASA and dSASA with 5 or 7 cal/(mol·Å<sup>2</sup>).

For Trp-cage, two production runs starting from unfolded and the first NMR structure were simulated for 1.0  $\mu\text{s}$  in GB, LCPO, pwSASA and dSASA, with a REMD ladder of 8 temperatures (247.7, 264.0, 281.4, 300.0, 319.8, 340.9, 363.3, 387.3 K). The backbone RMSD cutoff 2.0 Å was applied to calculate the fraction of folded. For homeodomain variant, two production runs starting from extended and the first NMR structure were simulated with a ladder of 10 temperatures (288.7, 300.0, 311.7, 323.9, 336.6, 349.8, 363.5, 377.7, 392.4, 407.8 K) for 2.0  $\mu\text{s}$  in GB and pwSASA, 1.0  $\mu\text{s}$  for LCPO and 0.7  $\mu\text{s}$  for dSASA. The backbone RMSD cutoff 5.0 Å was used to calculate the fraction of folded. These cutoffs are consistent with the study in.<sup>2,20</sup> MD simulations were not performed for the RNA systems, since weaknesses in current RNA force fields make it challenging to obtain stable simulations for stable hairpins such as 2KOC even with fully explicit water.<sup>45</sup>

## Results

### Molecular SASA estimation

In the test set to validate SASA estimation, we selected eight proteins from the set of the previously examined proteins for ab initio protein folding,<sup>2</sup> in which the set of proteins have diverse topologies: Trp-cage (20 residues), Fip35 (33 residues), NTL9 (39 residues), BBL (47 residues), NuG2variant (56 residues), Cspa (69 residues), Lambda-repressor (80

residues) and Top7 (92 residues). An ensemble of structures for every protein were extracted from the protein folding trajectories in that work with an even interval, obtaining a set of conformations with diverse atomic and molecular SASA values. The corresponding data were estimated for every conformation using ICOSA, LCPO and pwSASA and dSASA. The numerical method ICOSA can provide more accurate SASA estimation for molecules than LCPO and pwSASA, so we mainly compared the molecular SASA values from dSASA with ICOSA. The results are shown in Figure 6 and every point represents one conformation in the trajectory. In all the systems, dSASA can generate well-correlated SASA values with ICOSA as Pearson correlation coefficients of the linear regression are  $R^2 = 0.98, 0.99$  and the slopes are also close to 1. We note that this close agreement in fact demonstrates the relative accuracy of ICOSA as compared with our exact geometric method.

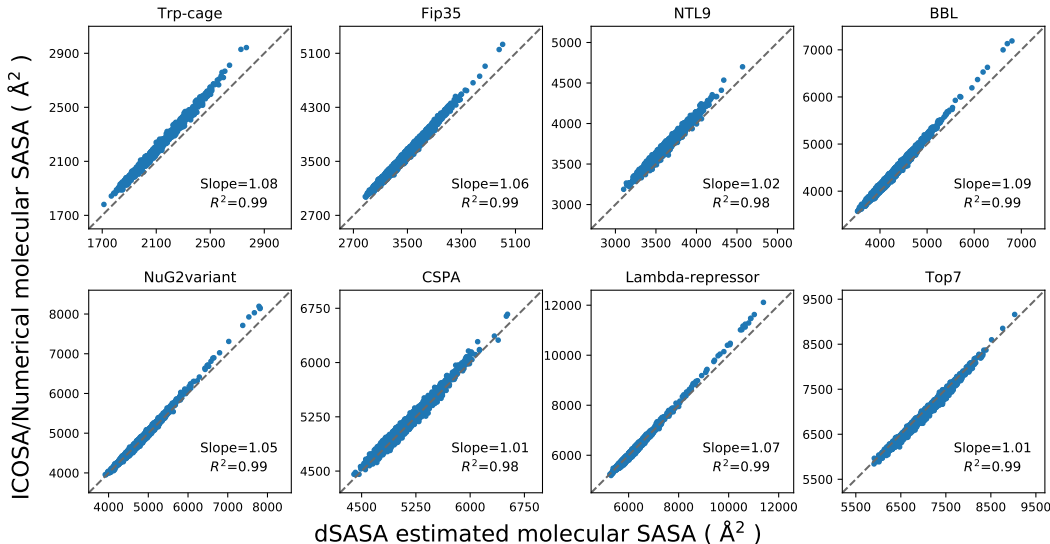


Figure 6: Comparison of molecular SASA between dSASA and ICOSA numerical method for the proteins. Each point represents one conformation of the protein. The diagonal dashed lines indicate perfect agreement.

The molecular SASA values from LCPO and pwSASA were calculated using the same conformations and compared with ICOSA method. As shown in Figure 7, the discrepancy of LCPO molecular SASA values with ICOSA values varies for different conformations in the systems. Overall, the LCPO tends to underestimate the values with most of the values are

above the perfect agreement line. The correlation coefficients are around 0.9 but it is lower than 0.9 for NTL9 and Cspa while the slopes are less than 0.9 for these two systems as well. The results became worse with pwSASA shown in Figure 8, where the similar phenomenon arose: the discrepancy of pwSASA molecular SASA values with ICOSA values varies for conformations in the systems; furthermore, all the correlation coefficients are less than 0.9 and the coefficients can be less than 0.8 for NTL9, BBL, NuG2variant and Cspa.

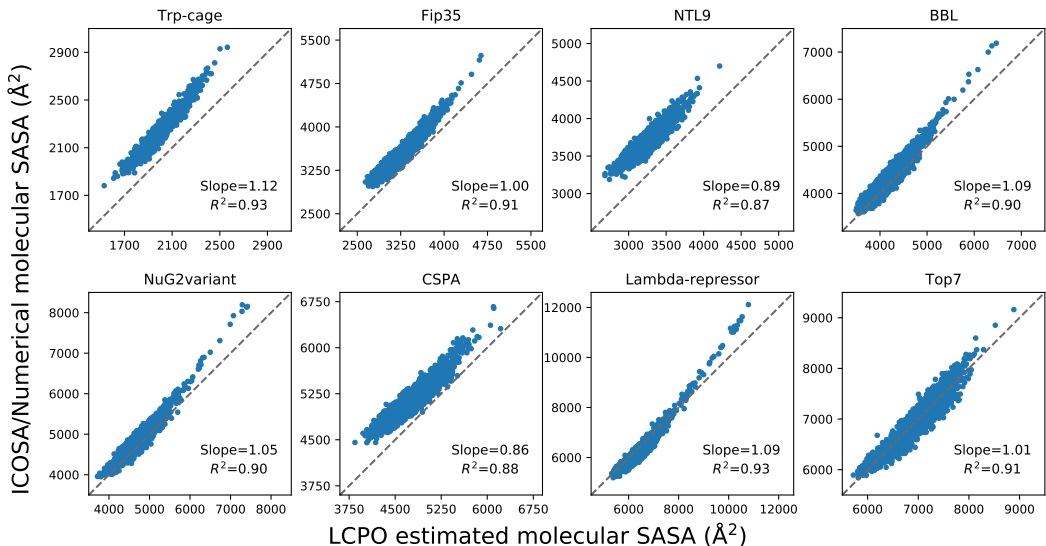


Figure 7: Comparison of molecular SASA between LCPO and ICOSA numerical method for the proteins. Each point represents one conformation of the protein. The diagonal dashed lines indicate perfect agreement.

Both LCPO and pwSASA use pairwise overlaps which can be relatively faster, but the inexact calculation will lead to inaccuracy especially for some types of atoms, for example both methods occasionally produce negative SASA values for individual atoms while the atomic values from dSASA are all nonnegative. LCPO includes the higher order correction terms to account for the overlap between two neighbors of the central atom. Then several atom types are defined by the environment, such as the atomic number, the number of bonded neighbors and the state of hybridization. Every atom type has predefined parameters through training. Because of less data for certain atom types, the parameter values for these atoms may generate unexpected values which explains the discrepancy for the proteins above. As

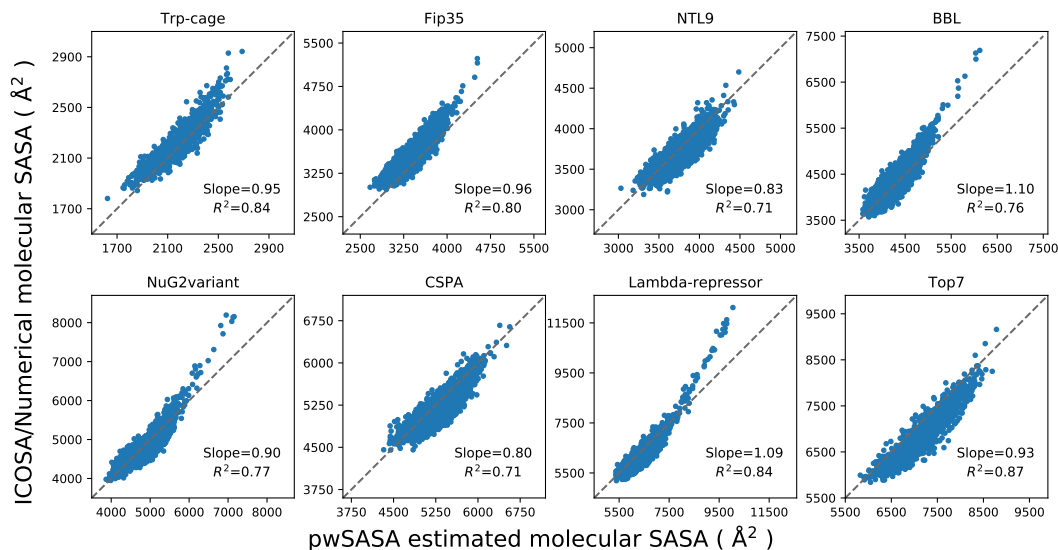


Figure 8: Comparison of molecular SASA between pwSASA and ICOSA numerical method for the proteins. Each point represents one conformation of the protein. The diagonal dashed lines indicate perfect agreement.

noticed in,<sup>28</sup> the method works better for more exposed atoms and tends to overestimate the surface area of some buried atoms, such as the carboxyl carbons of the amino acids Asp, Asn, Gln and Glu. pwSASA only considers the overlap between neighboring pairs and was trained based on a number of protein systems to match the values of LCPO. Even though introducing more atom types, it suffers similar problems as LCPO, likely due to insufficient training data and accounting for 3-body and higher terms only in an average way. In fact, the pwSASA values are not the original SASA values because it introduces an empirical adjustment to the SASA values to compensate for a systematic divergence from the numerical ICOSA calculations, which may adversely affect the overall accuracy as well.

## Speed Comparison in GB/SA MD

The original version of dSASA was implemented on CPUs, and the transformation to GPU version speeds up the computation by taking advantage of the parallel computing. The program written in CUDA was integrated to the Amber20 version. In sander, pmemd, or pmemd.cuda, setting the *gsa* flag to 4 in GB suite will activate GB/SA simulations with

dSASA. We first compared the speed of GB/SA MD on GPU with CPU and next with other methods using 10 proteins with various sizes ranging from 10 residues to 190 residues. 4 of them were selected from above and the PDB code of the rest are given: 10 (PDB code, 5AWL), 20, 33, 52 (2P6J), 69, 92, 110 (1BYW), 130 (1E6K), 162 (2B75) and 190 (1BK7).

We examined the method on the CPU Intel Platinum 8268 2.90GHz and on the GPU Nvidia RTX 2080 Ti. The estimated wallclock time for one step SASA calculation on CPU is from 12 ms (10 residues) to 90 ms (190 residues), while the time trend on GPU is from 12 ms to 30 ms. The wallclock time here and below is averaged over several simulations. The speed on CPU is comparable to the speed on GPU when the size of protein is small, and the speed on GPU becomes relatively faster as the size of the protein increases. The estimated wallclock time in hours required to complete 1 ns GB/SA simulation in Amber is given in the left of Figure 9. The wallclock time required to finish 1 ns simulation on the CPU increases from 0.87 hour (10 residues) to 33.80 hours (190 residues), while the corresponding trend of timing on the GPU is from 0.6 hour to 1.87 hours. When the size of protein is small, the improvement of GPU over CPU version is small with multiplicity 1.44 times for 10 residues, while the multiplicity grows as the size of molecule is increasing with 18.07 times for 190 residues.

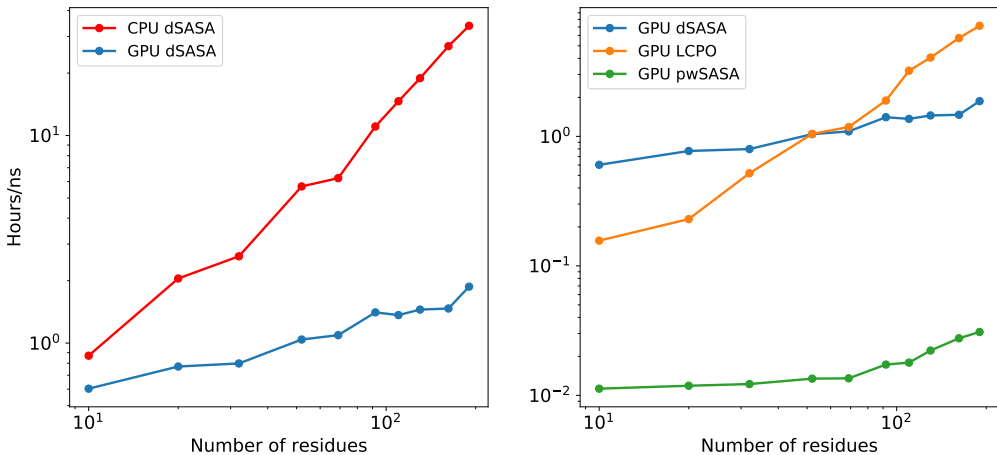


Figure 9: Left: Performance of dSASA for various sizes of proteins on CPU and GPU. The y-axis shows the wallclock time in hours to complete 1 ns simulation. Right: Performance of different methods for proteins on GPU.

We next compared the speed with other methods in Amber using the same set of systems above. The wallclock time in hours needed to complete 1 ns simulation is provided in the right of Fig. 9. The speed of pwSASA is stable and much faster than other two methods as it is designed to have a fast and simple estimation with more approximations, so we can confirm that the accuracy of the method is at cost of speed as LCPO and dSASA can reproduce more accurate SASA values by the testing above. As can be seen in Fig. 9, the speed of dSASA is more stable as the size of the system grows, while LCPO has a steeper increasing trend when the size of the molecule becomes larger. When the size of the system is small (10 residues), dSASA is slower than LCPO. While the crossing point is around the size of 69 residues and after that, dSASA starts to gain advantage over LCPO for 69, 92 residues and the multiplicity becomes greater for larger systems. This phenomenon fits the computation complexity for these two methods: the computation complexity of LCPO is  $O(N^2)$  while the complexity of dSASA is  $O(N \log N)$  (the complexity of tetrahedrization algorithm), so the system with a larger size will have relatively better performance. Another reason is that the GB/SA simulation with LCPO is CPU/GPU hybrid so the data transfer further hinders the performance. On the other hand, pwSASA is  $\sim 80$  times faster than dSASA for small proteins, reducing to  $\sim 50$  times for larger proteins. While pwSASA is a pairwise method and its computation complexity is also  $O(N^2)$ , its computation is efficiently embedded in the overall energy computation which is of the same order.

## Stability Analysis of Proteins

dSASA can estimate the molecular SASA accurately in proteins with diverse shapes based on the testing above. With the speedup on GPUs, it is now possible to examine its performance in GB/SA simulations and evaluate its effect on the stability of protein structures. Then we simulated two proteins with dSASA and compared the results with experiments, GB-only and other SASA methods. The simulation results are still dominated by the force field and the solvent model, so such comparisons should be used by caution as the inaccuracies in the

solute force field may affect the simulations and the solvent models may become less reliable away from 300 K. The trends in the simulations can provide useful information albeit with these limitations. The thermal stability profiles of Trp-cage and the fraction of folded on homeodomain are calculated from REMD at various temperatures. The temperature ladder for Trp-cage contains 8 replicas (247.7-387.3 K, see Methods for full list). The ladder for homeodomain has 10 replicas (288.7-407.8 K).

As investigated in,<sup>20</sup> the surface tension 5 cal/(mol·Å<sup>2</sup>) is working properly and is the default value in GB/SA in Amber, and larger value 7 cal/(mol·Å<sup>2</sup>) is expected to give more structures within native range and the value is 7.2 cal/(mol·Å<sup>2</sup>) as proposed in.<sup>5</sup> For Trp-cage system, we tested the surface tension 5 and 7 to check the performance of the method. For homeodomain, we only used the value 7, which can produce more near-native states in the trajectories.

The thermal stability profiles of Trp-cage with surface tension 5 and 7 cal/(mol·Å<sup>2</sup>) were first computed and the fractions of the near native conformations are given in Figure 10 for all methods. The left shows the results with the surface tension 5 cal/(mol·Å<sup>2</sup>), while the right shows the results with the surface tension 7 cal/(mol·Å<sup>2</sup>). At the surface tension 5, all three methods incorporating the nonpolar term can achieve better agreement with experiments by creating closer thermal stability profiles at various temperatures. At 300 K, dSASA can generate ~75% fraction of folded states while other methods with nonpolar term and experimental results are ~80% and the GB-only can produce ~50%. When the temperature is greater than 310 K, the fraction of folded from dSASA is still slightly smaller than other methods but dSASA can produce closer results to the experiments. The predicted melting temperature  $T_m$  is 316.8 K and the experimental result is around 317 K, which is better than the GB-only (predicted  $T_m$  is around 300 K) and other methods (predicted  $T_m$  is greater than 320 K for LCPO and pwSASA). Moreover, it is expected that a larger surface tension can generate more native-like conformations through which we can prove that the nonpolar term does not destroy the system. As shown in the right of Figure 10, all three methods

can generate more native-like structures with surface tension  $7 \text{ cal}/(\text{mol} \cdot \text{\AA}^2)$ . To show the

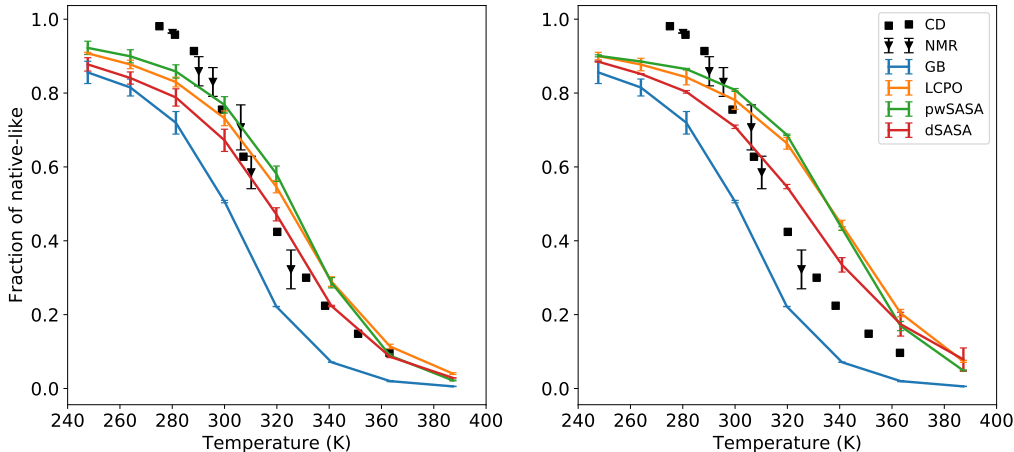


Figure 10: Thermal stability profiles for Trp-cage with surface tension  $5$  (left) and  $7 \text{ cal}/(\text{mol} \cdot \text{\AA}^2)$ (right) in GB and GB/SA simulations respectively, including experimental data.

changing process of RMSD, we provide the detailed trajectories at several temperatures for Trp-cage starting from extended and native states with surface tension  $5 \text{ cal}/(\text{mol} \cdot \text{\AA}^2)$  in Supporting Information.

For homeodomain variant, we first simulated REMD with surface tension  $5 \text{ cal}/(\text{mol} \cdot \text{\AA}^2)$  using pwSASA for  $2 \mu\text{s}$ , but we noticed that the fraction of folded was far from the experimental results at higher temperatures. The experimental measured  $T_m$  is greater than  $372 \text{ K}$ , so we expect a larger surface tension can achieve better agreement with experiments. We then simulated with surface tension  $7 \text{ cal}/(\text{mol} \cdot \text{\AA}^2)$  for all GB/SA methods. All three methods with nonpolar term can improve the fraction of folded at various temperatures compared to GB-only. The fractions of native-like starting from native and unfolded states at selected temperatures are shown in Figure 11. Since the experimental  $T_m$  is greater than  $372 \text{ K}$ , we expect that the system will not collapse at high temperatures. As two better methods, LCPO and dSASA can produce more native-like conformations at higher temperatures as shown in Figure 11. Moreover, dSASA generated more native-like conformations than LCPO at various temperatures, proving that our method can have a better effect on stabilizing the system. The detailed trajectories of homeodomain at selected temperatures starting from



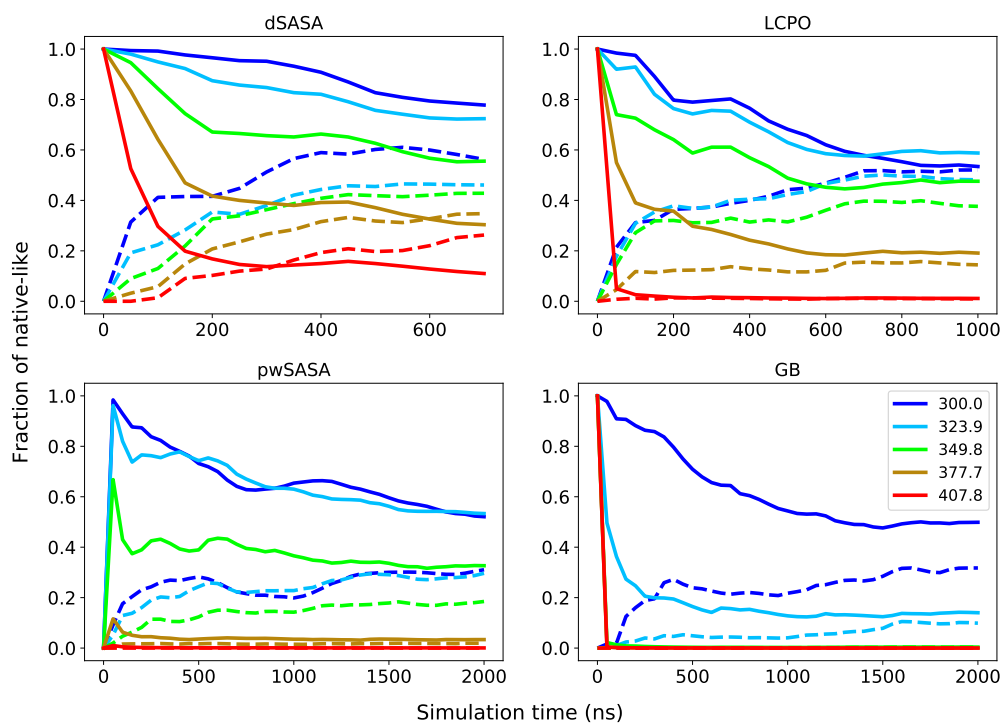


Figure 11: Fraction of native-like structures for homeodomain starting from native (solid lines) and extended (dashed lines) at 5 selected temperatures. Top left: dSASA; Top right: LCPO; Bottom left: pwSASA; Bottom right: GB-only.

extended and native states with surface tension  $7 \text{ cal}/(\text{mol} \cdot \text{\AA}^2)$  are given in Supporting Information. Furthermore, at the temperature 300 K, the difference between the native-like fractions of the simulations starting from native and extended is around 20% for dSASA and LCPO after 500 ns simulations, while that difference is around 40% for GB and pwSASA even after 1000 ns simulations, indicating that the inclusion of an accurate SASA term might be able to result in converging simulations at a faster rate. The improved performance from dSASA still has variance from the experimental results, for example the fraction of native-like at 377.7 K is around 35% while it is 50% in experiment. Such discrepancy may come from limitations of the SASA-based nonpolar term, or from the force field and the polar GB term, which are under active development.

We carried out another set of simulations for LCPO and dSASA, and the trajectories of fraction of folded are given in Supporting Information Figure 18. The simulation with dSASA can produce relatively less native-like conformations at lower temperatures in the trajectories starting from the native state while the fractions of folded starting from extended do not increase much at higher temperatures. On the other hand, in LCPO, the trajectories starting from native state produce less native-like conformations at lower temperatures than the trajectories starting from extended state. The results from both methods were not as expected because the MD simulations converge more slowly as the topological complexity of the system increases. Longer time simulations are clearly needed in such situation.

## Performance on RNA systems

Another advantage of dSASA over pwSASA is that it can be applied to other types of systems such as nucleic acids or small-molecule ligands, because only the radii of the atoms are considered without defining diverse atom types. Here we test this aspect using two RNA systems, 2KOC and 1ESH, and compare the molecular SASA values with ICOSA and other methods to see if the excellent performance on proteins is transferable to other biopolymers.

To validate the SASA estimation, we simulated the systems with GB-only method for

1  $\mu$ s and extracted  $10^4$  conformations for each system respectively. The comparison of the molecular SASA values between all methods is shown in Figure 12. dSASA can reproduce SASA values from ICOSA with more than 98% accuracy for both systems while LCPO and pwSASA produced divergent results for two systems and they had overall underestimation of the SASA values. The SASA estimates from pwSASA have large discrepancy from ICOSA method because it was trained exclusively on protein data so that the specific atom types in RNA were not represented properly by their definitions and the empirical adjustment is inadequate.

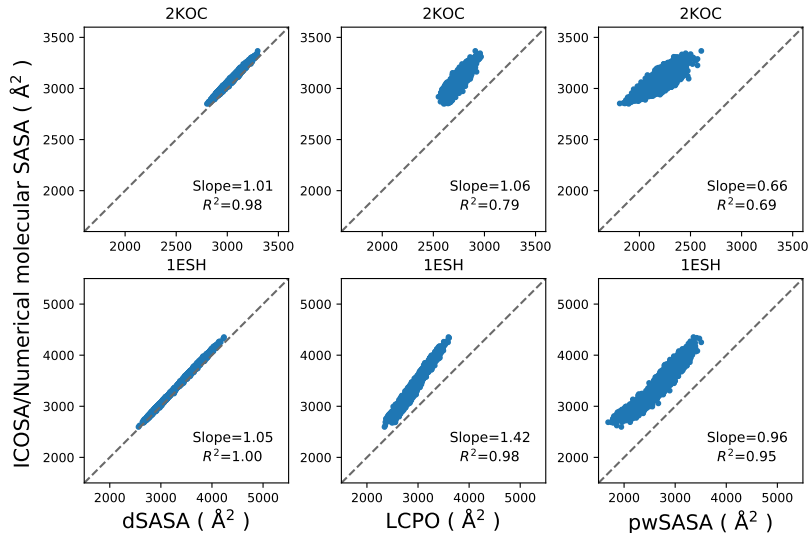


Figure 12: Comparison of molecular SASA values from various methods; each point represents one conformation of the RNA. The first row is for 2KOC and the second is for 1ESH. The diagonal dashed lines indicate perfect agreement.

## Conclusions

In this work, we present dSASA, an analytical SASA evaluation method for molecules and its implementation on GPUs. In this approach, the weighted Delaunay tetrahedrization is first computed with the atomic coordinates and radii. Next the atomic and molecular SASA values are estimated using inclusion-exclusion method based on exterior simplices from tetrahedrization, resulting in more accurate estimation than LCPO and pwSASA in

reproducing numerical ICOSA SASA values. The pre-trained parameters for atoms in LCPO and pwSASA can speed up the calculation, but the data sometimes will provide biased values for certain types of atoms so the computed molecular SASA values had smaller correlation coefficients with the values from ICOSA method, such as in the protein systems Cspa and Nug2variant and RNAs. The estimation from dSASA can reproduce ICOSA values with correlation coefficients greater than 98%. Moreover, the current GPU version speeds up the calculation substantially compared with the CPU version, making it applicable for longer time GB/SA MD simulations especially for larger systems. The gain of speed arises from the complexity of the algorithm itself which will scales well for larger systems and from the full implementation on GPU devices removing the data transfer with CPUs. dSASA had a stable trend of speed as the size of molecule increases but it was still relatively slower than pwSASA. However, dSASA started to outperform LCPO when the system contains around 70 residues, and the performance for larger molecules would be more stable than LCPO because the computation in LCPO is pairwise and its implementation is CPU/GPU hybrid in Amber.

In the GB/SA simulations, two proteins (Trp-cage and homeodomain) were simulated and compared to other methods and experimental results. dSASA achieved comparable performance with LCPO and pwSASA on the small protein, and the performance became better for the larger system. The simulated melting curve with nonpolar term for Trp-cage was more consistent with the experimental measures compared with that without nonpolar term. In homeodomain variant, the melting temperature is greater than 372 K so we expect the system can maintain some amount of native-like conformations at higher temperatures. The simulations generated more extended conformations with other methods at high temperatures while dSASA can produce more native-like conformations in the trajectories which is closer to the experimental data.

As the program has been rigorously examined, we anticipate to extend its applications to other functions in Amber software, such as post-processing of the trajectories. Given the

accuracy of dSASA, the calculation of atomic and molecular SASA values can be benchmark data set for the training of the parameters in pwSASA approach for RNAs in the future, which will combine the advantage of its speed on GPUs and the accurate calculation from dSASA. The surface tension was assumed to be identical for all atoms in the implementation. As the SASA value is exact for every atom in dSASA, we can attempt to assign different surface tension values to atoms according to the atomic environment which is believed to be able to achieve more accurate solvation free energies. Furthermore, the calculation of SASA is based on the diagram of Laguerre intersection cells, so it can be easily extended to the computation of molecular volumes along with the corresponding atomic derivatives. Inclusion of the volume term will provide a more complete description for the nonpolar solvation term. With the volume derivatives, we can further examine the impact of the term on the stability of molecules in the MD simulations. Moreover, the current algorithm being used for weighted Delaunay Tetrahedrization, gReg3D, is designed to work on a set of random points and the size of the workspace depends on the distribution of points in the workspace. As it consumes nearly 70% of wallclock time of our surface area calculation, improvement on this algorithm will further speed up the simulations. The spatial information of the atoms and the molecular properties in Amber were not fully utilized, then the influence of including such information among atoms could be further explored. The program of dSASA written in CUDA will be freely available from the authors.

## Acknowledgement

This work was partly supported by NIH Grant RM1 GM135136 and by NSF Grant DMS 2054251. We gratefully acknowledge support from Laufer Center for Physical and Quantitative Biology.

## References

- (1) Fennell, C. J.; Dill, K. A. *Journal of Statistical Physics* **2011**, *145*, 209–226.
- (2) Nguyen, H.; Maier, J.; Huang, H.; Perrone, V.; Simmerling, C. *Journal of the American Chemical Society* **2014**, *136*, 13959–13962.
- (3) Perez, A.; Morrone, J. A.; Brini, E.; MacCallum, J. L.; Dill, K. A. *Science Advances* **2016**, *2*, 1601274.
- (4) Stikoff, D.; Sharp, K. A.; Honig, B. *The Journal of Physical Chemistry* **1994**, *98*, 1978–1988.
- (5) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. *Journal of the American Chemical Society* **1990**, *112*, 6127–6129.
- (6) Kollman, P. A.; Massova, I.; Reyes, C.; Kuhn, B.; Huo, S.; Chong, L.; Lee, M.; Lee, T.; Duan, Y.; Wang, W.; Donini, O.; Cieplak, P.; Srinivasan, J.; Case, D. A.; Cheatham, T. E. *Accounts of Chemical Research* **2000**, *33*, 889–897.
- (7) Levy, R. M.; Zhang, L. Y.; Gallicchio, E.; Felts, A. K. *Journal of the American Chemical Society* **2003**, *125*, 9523–9530.
- (8) Honig, B.; Nicholls, A. *Science* **1995**, *268*, 1144–1149.
- (9) Eisenberg, D.; McLachlan, A. D. *Nature* **1986**, *319*, 199–203.
- (10) Wagoner, J. A.; Baker, N. A. *Proceedings of the National Academy of Sciences* **2006**, *103*, 8331–8336.
- (11) Allison, J. R.; Boguslawski, K.; Fraternali, F.; van Gunsteren, W. F. *The Journal of Physical Chemistry B* **2011**, *115*, 4547–4557.
- (12) S, S. G.; Ryde, U. *Expert Opinion on Drug Discovery* **2015**, *10*, 449–461.

- (13) Onufriev, A.; Bashford, D.; Case, D. *Proteins: Structure, Function, and Bioinformatics* **2004**, *55*, 383–394.
- (14) Mongan, J.; Simmerling, C.; McCammon, J. A.; Case, D. A.; Onufriev, A. *Journal of Chemical Theory and Computation* **2007**, *3*, 156–169.
- (15) Nguyen, H.; Roe, D. R.; Simmerling, C. *Journal of Chemical Theory and Computation* **2013**, *9*, 2020–2034.
- (16) Nguyen, H.; Perez, A.; Bermeo, S.; Simmerling, C. *Journal of Chemical Theory and Computation* **2015**, *11*, 3714–3728.
- (17) Case, D. A. et al. *University of California: San Francisco* **2017**,
- (18) Hawkins, G. D.; Cramer, C. J.; Truhlar, D. G. *Chemical Physics Letters* **1995**, *246*, 122–129.
- (19) Lang, E. J.; Baker, E. G.; Woolfson, D. N.; Mulholland, A. J. *Journal of Chemical Theory and Computation* **2022**, *18*, 4070–4076.
- (20) Huang, H.; Simmerling, C. *Journal of Chemical Theory and Computation* **2018**, *14*, 5797–5814.
- (21) Lee, B.; Richards, F. M. *Journal of Molecular Biology* **1971**, *55*, 379–400.
- (22) Connolly, M. L. *Journal of Applied Crystallography* **1983**, *16*, 548–558.
- (23) Durham, E.; Dorr, B.; Woetzel, N.; Staritzbichler, R.; Meiler, J. *Journal of Molecular Modeling* **2009**, *15*, 1093–1108.
- (24) Juba, D.; Varshney, A. *Journal of Molecular Graphics and Modelling* **2008**, *27*, 82–87.
- (25) Dynerman, D.; Butzlaff, E.; Mitchell, J. C. *Journal of Computational Biology* **2009**, *16*, 523–537.

- (26) Lee, K. H.; Chen, J. *Journal of Computational Chemistry* **2017**, *38*, 1332–1341.
- (27) Gong, X.; Chiricotto, M.; Liu, X.; Nordquist, E.; Feig, M.; Brooks III, C. L.; Chen, J. *Journal of Computational Chemistry* **2020**, *41*, 830–838.
- (28) Weiser, J.; Shenkin, P. S.; Still, W. C. *Journal of Computational Chemistry* **1999**, *20*, 217–230.
- (29) Gotz, A. W.; Williamson, M. J.; Xu, D.; Poole, D.; LeGrand, S.; Walker, R. C. *Journal of Chemical Theory and Computation* **2012**, *8*, 1542–1555.
- (30) Edelsbrunner, H. *Discrete & Computational Geometry* **1995**, *13*, 415–440.
- (31) Cazals, F.; Kanhere, H.; Lorient, S. *ACM Transactions on Mathematical Software (TOMS)* **2011**, *38*, 1–20.
- (32) Bryant, R.; Edelsbrunner, H.; Koehl, P.; Levitt, M. *Discrete & Computational Geometry* **2004**, *32*, 293–308.
- (33) Hummel, M. H. Delaunay-Laguerre geometry for macromolecular modeling and implicit solvation. Ph.D. thesis, The University of New Mexico, 2014.
- (34) Hummel, M. H.; Yu, B.; Simmerling, C.; Coutsiaris, E. A. *International Journal of Computational Geometry and Applications* **2018**, *28*, 1–38.
- (35) Koehl, P.; Akopyan, A.; Edelsbrunner, H. *Journal of Chemical Information and Modeling* **2023**, *63*, 973–985.
- (36) Shrake, A.; Rupley, J. A. *Journal of Molecular Biology* **1973**, *79*, 351–371.
- (37) Vasilyev, V.; Purisima, E. O. *Journal of Computational Chemistry* **2002**, *23*, 737–745.
- (38) Nanjappa, A. Delaunay Triangulation in R<sup>3</sup> on the GPU. Ph.D. thesis, Singapore National University, 2012.



- (39) Neidigh, J. W.; Fesinmeyer, R. M.; Andersen, N. H. *Nature Structural Biology* **2002**, *9*, 425–430.
- (40) Shah, P. S.; Hom, G. K.; Lassila, J. K.; Crowhurst, K. A.; Mayo, S. L. *Journal of Molecular Biology* **2007**, *371*, 1–6.
- (41) Nozinovic, S.; Fürtig, B.; Jonker, H. R. A.; Richter, C.; Schwalbe, H. *Nucleic Acids Research* **2010**, *38*, 683–694.
- (42) Kim, C. H.; Kao, C. C.; Tinoco, J. I. *Nature Structural Biology* **2000**, *7*, 415–423.
- (43) Hopkins, C. W.; LeGrand, S.; Walker, R. C.; Roitberg, A. E. *Journal of Chemical Theory and Computation* **2015**, *11*, 1864–1874.
- (44) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. *Journal of Chemical Theory and Computation* **2015**, *11*, 3696–3713.
- (45) Mráziková, K.; Mlýnský, V.; Kührová, P.; Pokorná, P.; Kruse, H.; Krepl, M.; Otyepka, M.; Banáš, P.; Šponer, J. *Journal of Chemical Theory and Computation* **2020**, *16*, 7601–7617.

## Supporting Information Available

### Energy conservation in MD

To examine the conservation of the energy and force stability, we carried out  $\sim 3.5$  ns of constant energy MD simulations using the protein HP36 (which has 36 residues) at 300K starting from the equilibrated NMR structure. The time step was 0.5 fs and SHAKE constraints were removed. Mixed precision GPU version (pmemd.cuda.SPFP) of Amber 20 and our program were used for GB and dSASA GB/SA simulations, respectively. For GB/SA simulations, surface tensions varied from 5, 10 and 20 cal/(mol $\cdot$ Å<sup>2</sup>). The energy terms were

printed out every 100 steps. The total energy deviations from the respective starting point were recorded to evaluate the energy conservation and force stability of the GPU code with and without the SASA term. As shown in Figure 13 and Table 1, the inclusion of dSASA produces comparable energy drift and standard deviation (SD) in the total energy. We can conclude that inclusion of the nonpolar term does not result in additional significant force instability.

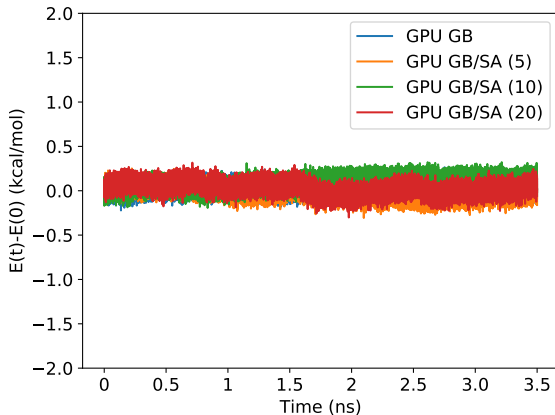


Figure 13: Energy deviation in constant energy simulations for HP36 from initial energy for GB and dSASA GB/SA with different surface tension parameters.

Table 1: The averages, the average deviations and standard deviations of the total energies in constant energy simulations for HP36 with GB and dSASA GB/SA.

Surface tension cal/(mol·Å <sup>2</sup> )	Etot Mean (kcal/mol)	E(t)-E(0) Mean (kcal/mol)	SD (kcal/mol)
0	-397.46215	0.0404465	0.05769035
5	-380.55516	-0.0185640	0.07033996
10	-363.58555	-0.0903457	0.06883961
20	-329.35631	-0.0302929	0.07151965

# Trajectories of proteins in REMD

In the main text, we showed the computed thermal stability profiles for Trp-cage and the fraction of folded for homeodomain from REMD, here we provide the detailed trajectories for Trp-cage Fig. 14, Fig. 15 and for homeodomain Fig. 16, Fig. 17 at various temperatures.

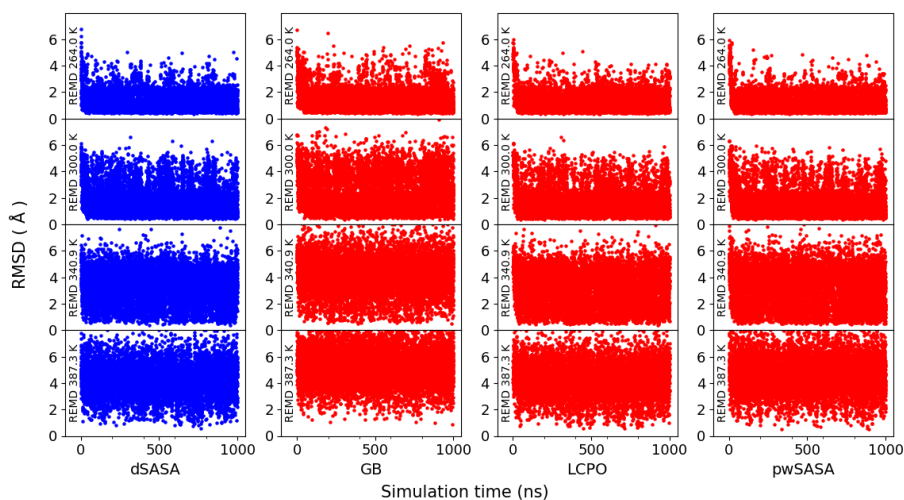


Figure 14: Trajectories at selected temperatures starting from extended state of Trp-cage with surface tension  $5 \text{ cal}/(\text{mol} \cdot \text{Å}^2)$  in GB and GBSA simulations.

## Fraction of folded for the second test of homeodomain

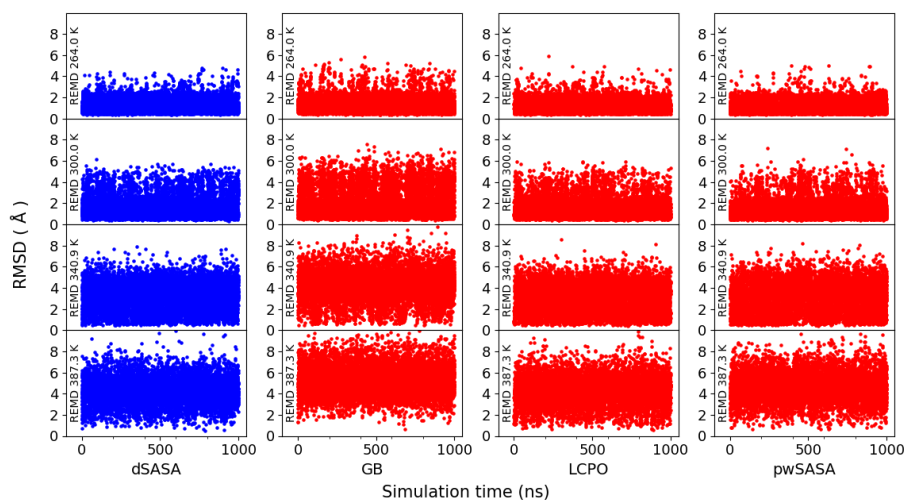


Figure 15: Trajectories at selected temperatures starting from native state of Trp-cage with surface tension  $5 \text{ cal}/(\text{mol} \cdot \text{Å}^2)$  in GB and GBSA simulations.

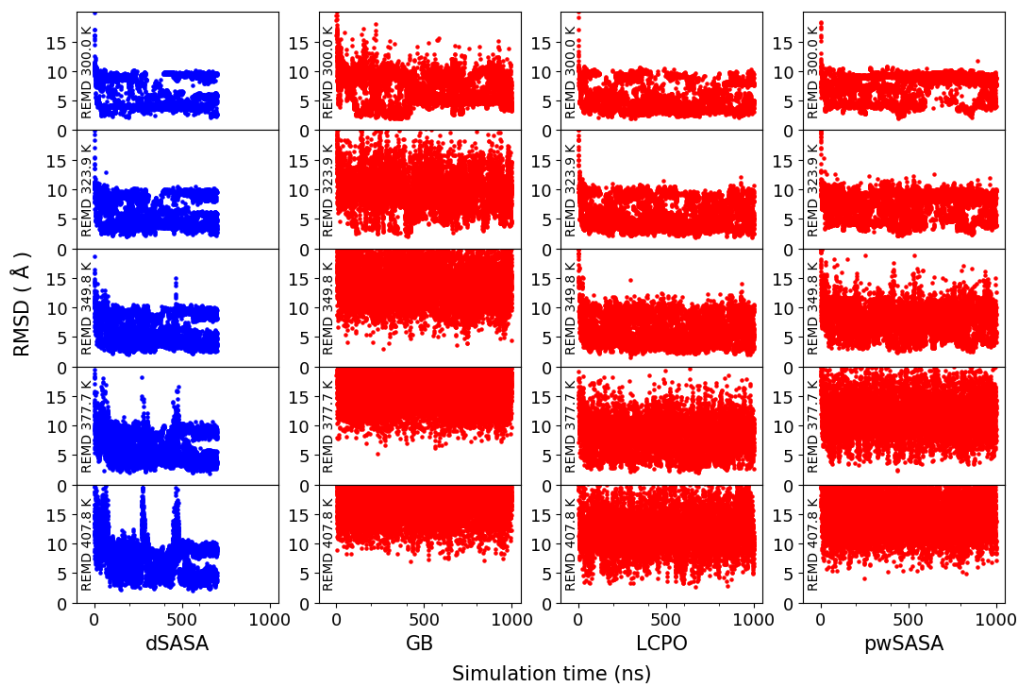


Figure 16: Trajectories at selected temperatures starting from unfolded state of home-domain with surface tension  $7 \text{ cal}/(\text{mol} \cdot \text{Å}^2)$  in GB and GBSA simulations.

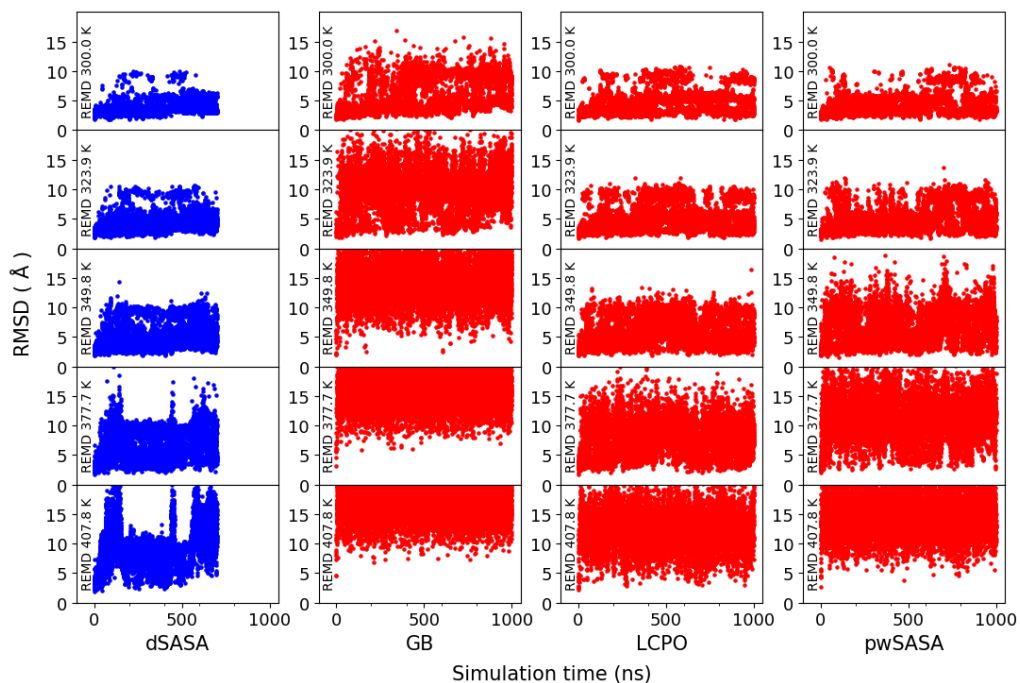


Figure 17: Trajectories at selected temperatures starting from native state of homeodomain with surface tension  $7 \text{ cal}/(\text{mol} \cdot \text{Å}^2)$  in GB and GBSA simulations.

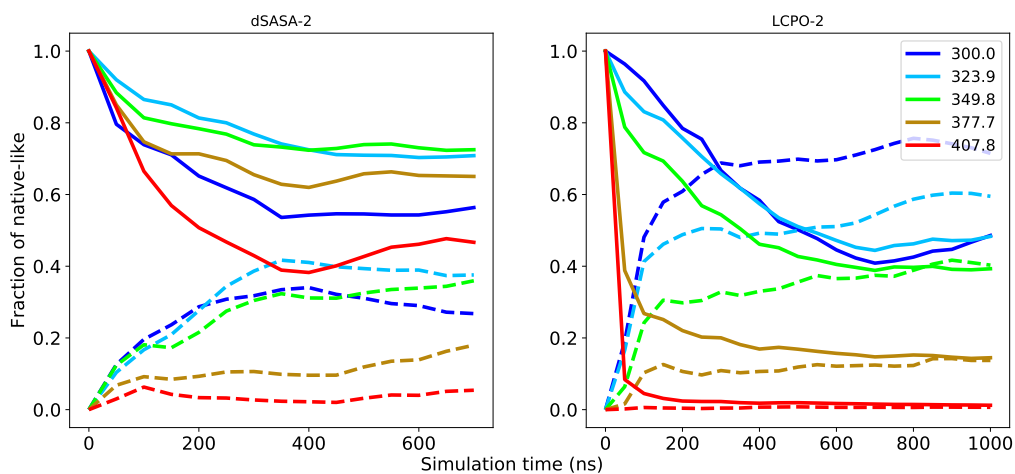


Figure 18: Fraction of native-like for homeodomain starting from native (solid lines) and extended (dashed lines) at 5 selected temperatures. Left: dSASA. Right: LCPO.