

High-Performance Analysis of Biomolecular Containers to Measure Small-Molecule Transport, Transbilayer Lipid Diffusion, and Protein Cavities

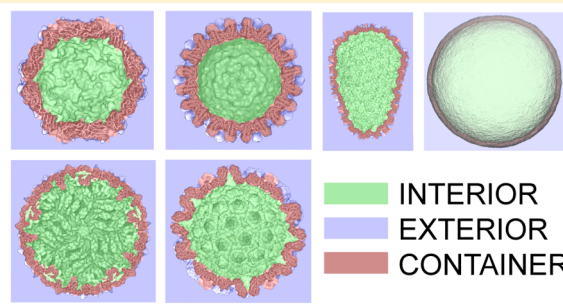
Alexander J. Bryer,^{†,§} Jodi A. Hadden-Perilla,^{†,§} John E. Stone,[‡] and Juan R. Perilla^{*,†}

[†]Department of Chemistry and Biochemistry, University of Delaware, Newark, Delaware 19716, United States

[‡]Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana–Champaign, Urbana, Illinois 61801, United States

Supporting Information

ABSTRACT: Compartmentalization is a central theme in biology. Cells are composed of numerous membrane-enclosed structures, evolved to facilitate specific biochemical processes; viruses act as containers of genetic material, optimized to drive infection. Molecular dynamics simulations provide a mechanism to study biomolecular containers and the influence they exert on their environments; however, trajectory analysis software generally lacks knowledge of container interior versus exterior. Further, many relevant container analyses involve large-scale particle tracking endeavors, which may become computationally prohibitive with increasing system size. Here, a novel method based on 3-D ray casting is presented, which rapidly classifies the space surrounding biomolecular containers of arbitrary shape, enabling fast determination of the identities and counts of particles (e.g., solvent molecules) found inside and outside. The method is broadly applicable to the study of containers and enables high-performance characterization of properties such as solvent density, small-molecule transport, transbilayer lipid diffusion, and topology of protein cavities. The method is implemented in VMD, a widely used simulation analysis tool that supports personal computers, clouds, and parallel supercomputers, including ORNL's Summit and Titan and NCSA's Blue Waters, where the method can be employed to efficiently analyze trajectories encompassing millions of particles. The ability to rapidly characterize the spatial relationships of particles relative to a biomolecular container over many trajectory frames, irrespective of large particle counts, enables analysis of containers on a scale that was previously unfeasible, at a level of accuracy that was previously unattainable.



INTRODUCTION

Living cells are complex containers, within which numerous reactions and biochemical processes must be carried out simultaneously to support life. Eukaryotes partition their intracellular space with subcontainers, membrane-enclosed structures including organelles and vesicles, allowing them to organize and regulate the various aspects of their metabolism. Viruses are also containers, packaging their genomes and accessory enzymes in protein shells called capsids, which may be further wrapped in lipid envelopes. Proteins themselves may include cavities embedded in their tertiary or quaternary structures, which can act as containers for ligands or substrates that recognize and bind them.

Importantly, all biomolecular containers are highly specialized to facilitate the specific interaction, reaction, or process they enclose and exhibit key biophysical properties that enable them to carry out their functions. Protein cavities that serve as binding sites engender ligand affinity by offering suitable pocket size and shape complementarity. Vesicles and capsids often play essential roles in maintaining the biochemical environments necessary for their compartmentalized processes,

regulating the concentration, pH, or pressure of their interior space by controlling the passage of small molecules, ions, protons, or solvent. Some membrane-bound structures must establish or preserve asymmetry—differing lipid compositions across the inner and outer leaflets of the bilayer—to drive cellular functions, requiring transbilayer diffusion and protein-catalyzed translocation of lipids between leaflets.^{1–3}

Molecular dynamics (MD) simulations are now routinely applied to study large biological systems, including containers such as virus capsids and viral envelopes, at both coarse-grained and all-atom levels of detail.^{4–6} Large-scale simulation studies of lipid vesicles have also become common.^{7–12} Because MD simulations can be used to characterize the influence of containers on their surrounding biochemical environments at the resolution of individual particles—a level of detail inaccessible to experimental methods—the technique provides a powerful approach to investigate the properties of containers that underlie their functions. For example, by

Received: April 18, 2019

Published: September 17, 2019

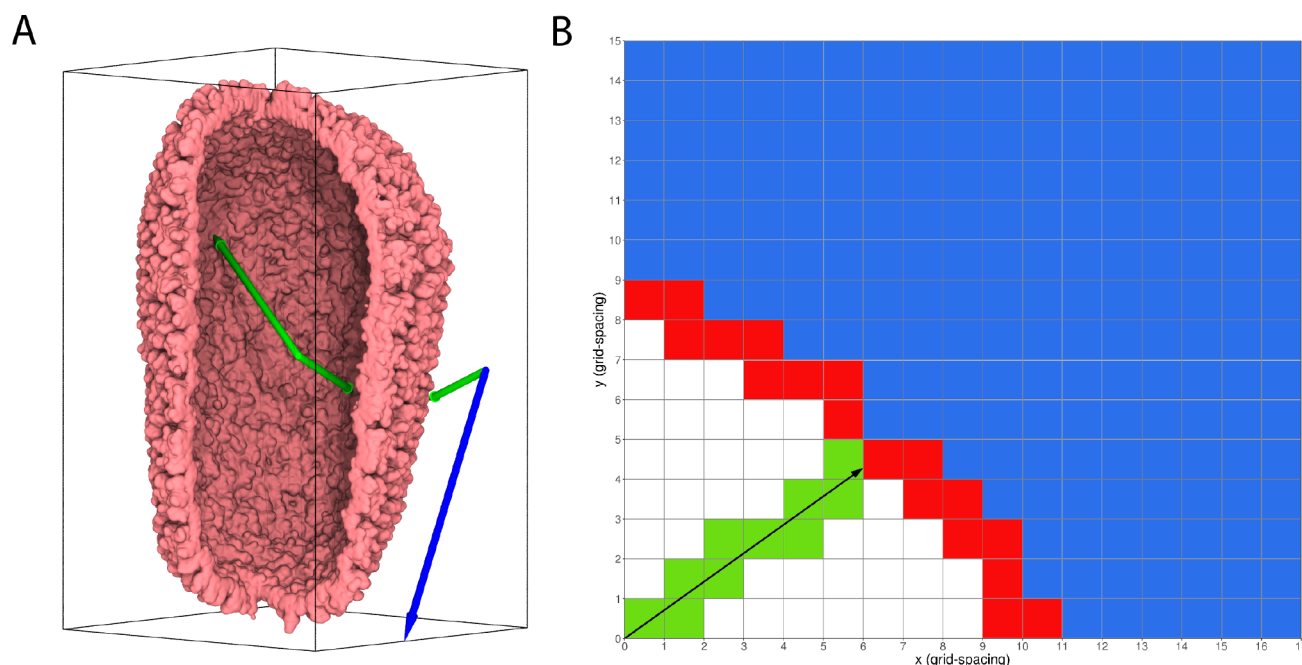


Figure 1. Ray casting and voxel traversal. (A) Schematic representation of *measure volinterior* applied to a cross section of the HIV-1 capsid. Here, $N_{\text{rays}} = 2$ rays are cast from each voxel in a grid set to the size of the system bounding box. The rays are randomly and uniformly distributed, according to the spherical distribution given in eq 1. For a given voxel shown on the capsid's exterior, one of the rays (blue) traverses the grid of voxels unimpeded over the entire interval up to the wall of the bounding box, indicating that the voxel is located outside the capsid container. For a given voxel shown on the capsid's interior, both of the rays (green) fail to strike the wall of the bounding box because they intersect (are blocked by) the capsid surface, indicating that the voxel is located inside the capsid container. (B) A 2-D version of the DDA algorithm. The ray, cast from the grid origin, traverses the grid visiting a sequence of cells (equivalent to voxels in 3-D) on its path to the system's bounding box. In this example case, the ray strikes voxels occupied by the container (red), determining that the ray originated inside the container (green) instead of outside (blue).

tracking the exchange of solvent and ions across the surface of virus capsids, researchers have been able to determine their small-molecule transport properties, revealing insights into the capsids' regulatory roles in reverse transcription and cellular signaling.^{13,14} Due to the size of virus capsids and most other containers of biological interest, however, simulated systems can encompass millions of particles, with those describing the solvent environment far outnumbering solute.

In the present article, a novel method is presented that enables rapid analysis of biomolecular containers, irrespective of large system particle counts. The method, implemented in VMD¹⁵ version 1.9.4 and referred to herein by its VMD command *measure volinterior*, automatically distinguishes container interior versus exterior, regardless of container shape, and detects the spatial relationships of particles relative to the container in a manner that is efficient and highly scalable. Instead of dealing with particles individually, *measure volinterior* uses 3-D ray casting to classify the space surrounding the container's molecular surface. VMD's atom selection feature then permits rapid collection of the identities of particles that are located inside or outside of the container. This information can be used to determine the number and density of particles within or around a container, characterize small-molecule transport by computing the exchange rates of particles flowing inward and outward across the container's surface, track transbilayer diffusion in lipid-based containers, and even identify and measure protein cavities that may represent small-molecule binding pockets or druggable sites.

In the following sections, details of *measure volinterior* theory and implementation are provided, along with guidelines for

selecting appropriate parameters to define the molecular surfaces of containers required for the method. Code examples are also included to demonstrate the application of *measure volinterior* to perform the container analyses listed above. Finally, the protocol for analyzing an MD simulation trajectory of a multimillion-atom biomolecular container is described, taking advantage of supercomputers such as ORNL's Titan and Summit and NCSA's Blue Waters, which include access to high-performance Lustre file systems.

METHODS

Theory and Implementation of *measure volinterior*. A tractable approach for analyzing MD simulations of biomolecular containers requires automated execution of the following in an efficient and highly scalable manner to enable rapid processing of millions of particles over thousands of trajectory frames: (i) recognition of the container, (ii) detection of container interior versus exterior, regardless of container shape or orientation, and (iii) classification of noncontainer particles according to their location inside or outside of the container. The *measure volinterior* approach addresses these challenges by using a 3-D ray casting method combined with a spatially discretized or *voxelized* system and the space around it. Ray casting has previously been proposed as a general and efficient strategy for characterizing properties of molecules that arise from their surface geometry.¹⁶ Importantly, *measure volinterior* can be applied to study containers of arbitrary shape, and the results it produces are spatially invariant. The algorithm is implemented in C++ in

VMD¹⁵ version 1.9.4, where ray-level data parallelism maximizes computational performance.

The VMD command for calling *measure volinterior* on a container described by atom selection *sel* in molecule *mol*, is given in Code Snippet 1. A given container is defined by its molecular surface, calculated using VMD's QuickSurf algorithm. The user must provide parameters for *Radius Scale*, *Isovalue*, and *Grid Spacing*. For accurate results, the parameters must yield a nonporous container, characterized by a continuous molecular surface without holes; otherwise, the user must enable fuzzy-boundary detection. See the following sections for guidelines for selecting QuickSurf parameters for various types of biomolecular containers, as well as details on dealing with fuzzy container boundaries. The user must also indicate the number of rays, N_{rays} . Upon calling *measure volinterior* on a container selection, a 3-D grid is generated with the same dimensions as the system bounding box for the given trajectory frame. For each voxel in the grid, n rays are cast outward. In the default *measure volinterior* implementation, the container interior versus exterior is determined based on whether the rays strike the system bounding box or are blocked by the container wall (Figure 1A).

Code Snippet 1 Calling *measure volinterior* on a container selection.

```
set Nvoxels [measure volinterior $sel -res [expr $RadiusScale / 0.2]
             -isovalue $Isovalue -spacing $GridSpacing -nrays $n -mol $mol]
```

Each ray is cast through a 3-D grid defined by the system bounding box, using the Digital Differential Analysis (DDA) algorithm.¹⁷ The DDA voxel traversal algorithm incrementally computes x , y , and z voxel indices, visiting all voxels pierced by the ray between its origin and a wall or obstacle. Using the DDA algorithm for voxel traversal, the rays are rasterized through the aforementioned voxelized space, sampling the voxel contents to check for container intersections, thereby allowing the starting voxel to be classified as interior or exterior. A 2-D version of the algorithm is illustrated in Figure 1B, depicting the process of voxel traversal by a ray and the resulting classification of voxel space.

In DDA, rays are cast outward from the origin of each voxel with directions following a random, uniform spherical distribution given by

$$\begin{aligned}x &= \cos \theta \sin \phi \\y &= \sin \theta \sin \phi \\z &= \cos \phi\end{aligned}\quad (1)$$

where

$$\begin{aligned}\theta &= 2\pi u \\ \phi &= \arccos(2v - 1)\end{aligned}$$

and u and v are uniformly distributed on $[0, 1]$, that is

$$u \sim v \sim U([0, 1])$$

At $N_{\text{rays}} \geq 32$, uniformity of ray directions is ensured via a Poisson disk sampling procedure.¹⁸ Example sets of ray direction vectors, plotted on the surface of a unit sphere, are shown in Figure S1B. Given uniformity of ray directions, the volume classification is invariant to choice of N_{rays} , as demonstrated in Figure S1A. The direction (x, y, z) is used to compute a step along the tracing ray, such to match the width of one voxel in each dimension, as follows

$$d\zeta = \begin{cases} \frac{1}{2\zeta} \sqrt{(x^2 + y^2 + z^2)} & \zeta < 0 \\ \frac{3}{2\zeta} \sqrt{(x^2 + y^2 + z^2)} & \zeta > 0 \\ 0 & \zeta = 0 \end{cases} \quad (2)$$

where

$$\zeta \in \{\hat{x}, \hat{y}, \hat{z}\} \equiv \mathbb{M}$$

and where \mathbb{M} is the set of all unit vectors.

With the differentials defined by eq 2, DDA determines digital steps of size m_ζ that propagate the ray in voxel space, one voxel index increment at a time. To accomplish the ray propagation, the algorithm first finds the minimum valued ζ of the current direction vectors. The minimum, or lagging dimension, is given a step m_ζ of ± 1 , and the other two dimensions are given a step of 0. Therefore, the ray propagates according to

$$\zeta_i = \zeta_{i-1} + m_\zeta \quad (3)$$

where $i = 1 \dots N$. N is determined conditionally based on whether the ray intersects voxels belonging to the molecular surface or strikes the bounding box.

If any one of the rays cast from a voxel strikes the bounding box, then that voxel is considered to be located on the container's exterior and marked with a grid value of 5.0 (Figure 1A, blue arrow). Conversely, if none of the rays cast from a voxel strike the bounding box, because they are all blocked by the inner walls of the container, then that voxel is considered to be located on the container's interior and marked with a grid value of 0.0 (Figure 1A, green arrows). Voxels that coincide spatially with the container surface and represent the boundary between interior and exterior are considered container voxels, marked with a grid value of -5.0 . Importantly, because the system bounding box is used as an essential landmark of exterior space, the container system must be wrapped to its unit cell and not split across a periodic boundary.

The output of *measure volinterior* is a list containing the total voxel count and the number of voxels identified as inside, outside, or comprising the container, as well as a 3-D volume grid (e.g., vol0). Voxels in the grid are assigned with values indicating their interior ($\text{vol0} < 1$ and $\text{vol0} > -1$), exterior ($\text{vol0} > 0$), or boundary ($\text{vol0} < 0$) classification for the processed trajectory frame. The numbers of voxels identified in each region can be accessed via their list indices, as shown in Code Snippet 2. Further, by invoking the corresponding values from the 3-D grid, VMD's atom selection feature allows for fast region classification of system particles according to their location. Particles that happen to fall outside of the system bounding box, whose associated space is not detected by *measure volinterior*, likely also fall on the container exterior, and their grid values can be selected as those that are not a number ($\text{vol0} > 1$ and $\text{vol0} < -1$). Importantly, the relational operators used for atom selections should be written as intervals instead of equalities (e.g., $\text{vol0} < 1$ and $\text{vol0} > -1$ instead of $\text{vol0} = 0$) to ensure platform independent behavior. Code Snippet 3 shows how to define macros for classified regions of the output volume grid, for use in atom selections.

Code Snippet 2 Accessing indexed voxel counts returned by *measure volinterior*.

```
set VoxelsTotal [!index $Nvoxels 0]
set VoxelsOut [!index $Nvoxels 1]
set VoxelsIn [!index $Nvoxels 2]
set VoxelsBound [!index $Nvoxels 3]
```

Code Snippet 3 Assigning atom selection macros for use with *measure volinterior*.

```
atomselect macro vol0interior "vol0 < 1 and vol0 > -1"
atomselect macro vol0exterior "vol0 > 0"
atomselect macro vol0boundary "vol0 < 0"
atomselect macro vol0undefined "vol0 > 1 and vol0 < -1"
```

VMD 1.9.4, which includes full *measure volinterior* functionality, is compiled on ORNL's Titan and Summit and NCSA's Blue Waters supercomputers and is readily available to any researcher with access to these resources for postprocessing analysis of large MD simulation trajectories. For smaller systems, the method also performs well on desktop workstations. See the [Results and Discussion](#) section for examples demonstrating the usage of *measure volinterior* to analyze biomolecular containers for properties such as enclosed solvent density, small-molecule transport, and transbilayer lipid diffusion, as well as to detect and measure cavities within or between constituent proteins.

Selection of Parameters to Define Container Surfaces. When using *measure volinterior*, containers are defined by their molecular surfaces, as calculated by VMD's QuickSurf algorithm. QuickSurf generates a molecular surface extracted from a volumetric Gaussian density map, computed by summing the Gaussian densities from the particles in a region surrounding each grid point.^{19,20} User-defined parameters allow manipulation of the molecular surface representation to alter spatial fidelity and level of detail. For the fixed-boundary implementation of *measure volinterior*, accurate characterization of a container and its biophysical properties depends on selecting QuickSurf parameters that yield a continuous molecular surface with no holes. Otherwise, fuzzy-boundary detection must be enabled (see below). In the fixed-boundary case, surface defects and indentations do not impact the performance of *measure volinterior*, as long as they do not penetrate the surface completely. Overall, the container surface must be nonporous, such that rays cast from inside the container do not inadvertently escape to the container exterior, resulting in incorrect classification of the surrounding space. This means that containers with natural pores or channels, such as virus capsids, require QuickSurf parameters designed to mask such openings without sacrificing other important features of the container surface.

Three QuickSurf parameters are required by *measure volinterior* to define molecular surfaces, namely, *Radius Scale*, *Isovalue*, and *Grid Spacing*. *Radius Scale* is a radius scaling factor applied to all particles prior to computing their contributions to the volumetric density map; its value relates to overall spatial resolution of the final surface. A larger value of *Radius Scale* will produce a bulkier molecular surface, which may be needed to mask pores and prevent the escape of rays to the container exterior. *Isovalue* is the isovalue used when extracting the molecular surface from the density map and represents the scalar density with which the surface is generated. *Grid Spacing* is the distance between grid points, which defines the size and number of voxels. Smaller values of *Grid Spacing* will resolve finer details in the molecular surface; a value of 1 Å is sufficient to represent the surface at atomic detail. For larger container systems, it may be desirable to use larger values of *Grid Spacing*, sacrificing surface resolution for computational efficiency. Since a lower resolution grid contains fewer voxels,

and N_{rays} are cast per voxel, larger values of *Grid Spacing* result in significantly fewer rays cast per processed trajectory frame. Further, a lower resolution grid entails a smaller memory footprint and may prevent memory overflow during analysis of a large container. Decreasing the number of rays can also improve computational efficiency. While $N_{\text{rays}} = 64$ will likely be sufficient to characterize a highly detailed molecular surface, $N_{\text{rays}} = 32$ represents a reasonable compromise for large container systems, especially if their surfaces are relatively featureless.

Selection of suitable inputs for *measure volinterior* is an empirical process. VMD's graphical interface should be used to experiment with and identify parameters for QuickSurf that capture the essential topology of the biomolecular container of interest. For use with the fixed-boundary implementation of *measure volinterior*, the parameters should yield a molecular surface that is continuous. The output of *measure volinterior* for a given trajectory frame should be visualized to assess the performance of a candidate parameter set prior to production analysis. [Figure 2](#) shows cross sections of several container systems, along with volume slices from 3-D grids produced by calling *measure volinterior* on them. The parameters used to generate the discrete volume grids, provided in [Table 1](#), capture significant detail in the containers' surface features, while successfully masking pores and channels to prevent the escape of rays. The resulting classification of space around the containers is highly accurate, with the delineations of interior versus exterior conforming closely to the molecular structure.

The effects of using parameters that are not well suited to produce accurate results with *measure volinterior* are illustrated in [Figure S2](#). Importantly, parameters selected for a given system must perform well for all frames of a simulation trajectory. Because pores and channels may fluctuate in diameter during the course of simulation, parameters selected for use with the fixed-boundary implementation of *measure volinterior* must produce a molecular surface sufficiently thick to mask openings over all sampled conformations. For example, QuickSurf parameters optimized for the experimental HBV capsid structure ([Figure 2B](#)) allow the escape of rays to the container exterior for some frames during MD simulation at 310 K¹⁴ ([Figure S2A](#), left); slight adjustments to the parameters produce a closed surface and ensure accuracy over all frames during production trajectory analysis ([Figure S2A](#), right). On the other hand, parameters that produce an overly bulky surface will capture less topological detail and also lead to more space surrounding the container being classified as part of the container itself. For example, QuickSurf parameters that resolve fine surface definition in the ZIKV E/M protein shell lead to highly accurate classification of the space surrounding the container ([Figure S2B](#), left); a bulkier surface leads to extra space adjacent to the container being counted as part of the boundary ([Figure S2B](#), right).

Detection of Fuzzy Container Boundaries. To analyze containers with discontinuous molecular surfaces, such as those that include pores or holes, or those that represent open cavities, *measure volinterior* is equipped with an algorithm for fuzzy-boundary detection. In the fixed-boundary *measure volinterior* implementation, if any ray cast from a voxel strikes the system bounding box, that voxel is classified as being outside of the container. The output grid contains discrete space assignments of inside, outside, or boundary between (i.e., the container itself). When fuzzy-boundary detection is enabled, *measure volinterior* assigns a probability to each

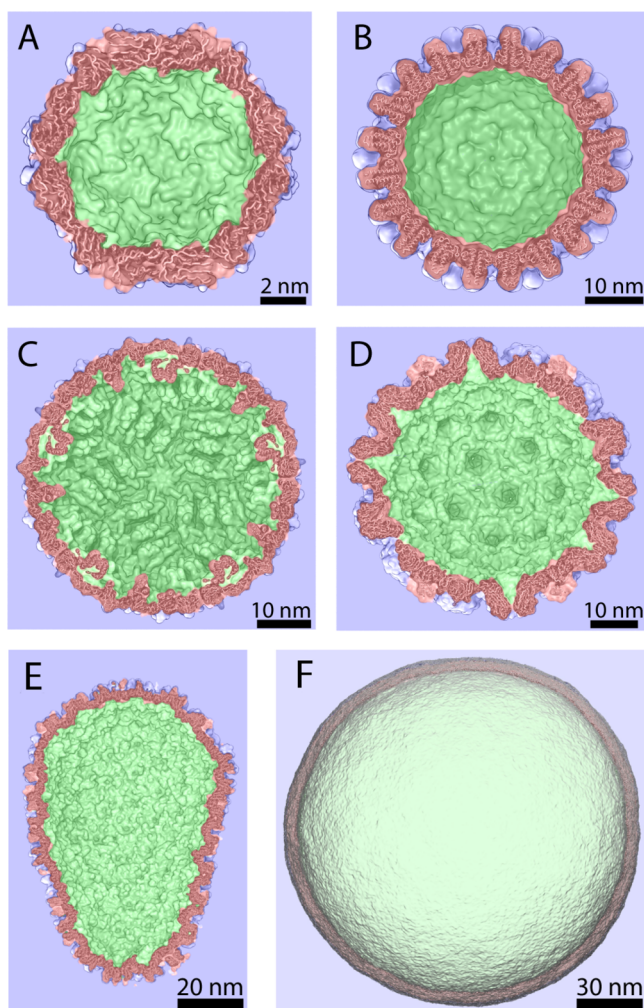


Figure 2. Visualization of outputs from *measure volinterior* with fixed-boundary detection. Six biomolecular container structures are shown in cross section, along with a volume slice from the 3-D volume grid produced by *measure volinterior* (e.g., vol0). Using the parameters listed in Table 1, space classification as interior (vol0interior, green), exterior (vol0exterior, blue), or boundary (vol0boundary, red) closely matches the molecular surface, capturing notable topological detail. See Code Snippet 3 to see how the macros specifying these regions of the grid, for example, vol0interior and vol0exterior, are defined. (A) satellite tobacco mosaic virus (STMV) capsid,²¹ (B) hepatitis B virus (HBV) capsid,²² (C) zika virus (ZIKV) envelope/membrane (E/M) protein shell,²³ (D) simian vacuolating virus 40 (SV40) capsid,²⁴ (E) HIV-1 capsid,²⁵ and (F) HIV-1 envelope membrane.

voxel based on the number of rays cast from it that fail to strike the bounding box. The probability describes, generally, the extent to which a voxel is occluded by the molecular surface or the degree to which it represents a cavity region. In that case, a

probability threshold or interval can be used to define the boundary between container interior and exterior, instead of relying on a closed shell to delineate these regions. Figure 3 shows cross sections of two container systems that contain pores or holes, along with volume slices from 3-D grids produced by calling *measure volinterior* on them with fuzzy-boundary detection enabled. Despite discontinuous molecular surfaces, the fuzzy-boundary detection method accurately identifies the interior versus exterior regions of the containers.

To enable *measure volinterior* for fuzzy-boundary detection, a 3-D grid \mathbb{T} containing N_{voxels} is used, where

$$N_{\text{voxels}} = N_x \cdot N_y \cdot N_z \quad (4)$$

Each of the N_{voxels} entries in \mathbb{T} stores a value τ_i , where $i \in [0, N_{\text{rays}}]$. A continuous, 3-D probability grid \mathbb{P} is then computed according to

$$\mathbb{P} = \frac{1}{N_{\text{rays}}} \cdot \mathbb{T} \quad (5)$$

where entries ρ_i in the grid range continuously between $[0, 1]$, and $\rho_i = 1$ describes a completely occluded voxel. The continuous probability grid \mathbb{P} may be further processed into a discrete volume grid, such as those output by default in the fixed-boundary implementation, using the *discretize* flag and a floating point argument between $[0, 1]$, which represents the probability threshold ρ_0 . Examples of what discretized grids look like, for different test cases and threshold values, are given in Figure S3B, D and Figure S1C. An example demonstrating all possible values of ρ_0 side-by-side is shown in Figure 5E.

The VMD command for calling *measure volinterior* with fuzzy-boundary detection is given in Code Snippet 4. By invoking values from the output grid (e.g., vol1), users can provide probability thresholds or intervals in their atom selections to define the boundary between container interior and exterior and classify particles according to their respective locations. Just as suitable QuickSurf parameters must be determined empirically for each container system based on visualization of *measure volinterior* results, the probability thresholds or intervals used with fuzzy-boundary detection must also be experimented with to determine what represents the ideal for each system prior to production analysis. Code Snippet 5 shows examples of how to define macros for regions of the continuous output grid \mathbb{P} , for use in atom selections; the examples pertain to a hypothetical System X, for which the given values were deemed appropriate following visual assessment.

Code Snippet 4 Calling *measure volinterior* with fuzzy-boundary detection.

```
set Nvoxels [measure volinterior $sel -res [expr $RadiusScale / 0.2]
-isoval $Isoval -spacing $GridSpacing -nrays $n -mol $mol -probmap]
```

Table 1. *measure volinterior* Parameters Used to Generate 3-D Grids Shown in Figure 2

Molecular structure	Diameter (nm)	Radius Scale (Å)	Isovalue	Grid Spacing (Å)	Rays, n
STMV capsid ²¹	17	1.2	0.8	0.8	64
HBV capsid ²²	35	1.7	0.8	0.9	64
ZIKV E/M protein shell ²³	47	1.8	0.9	1.5	64
SV40 capsid ²⁴	49	1.2	0.8	1.2	64
HIV-1 capsid ²⁵	109 × 66	1.8	0.9	1.5	32
HIV-1 envelope membrane	150	3.2	2.0	2.33	6

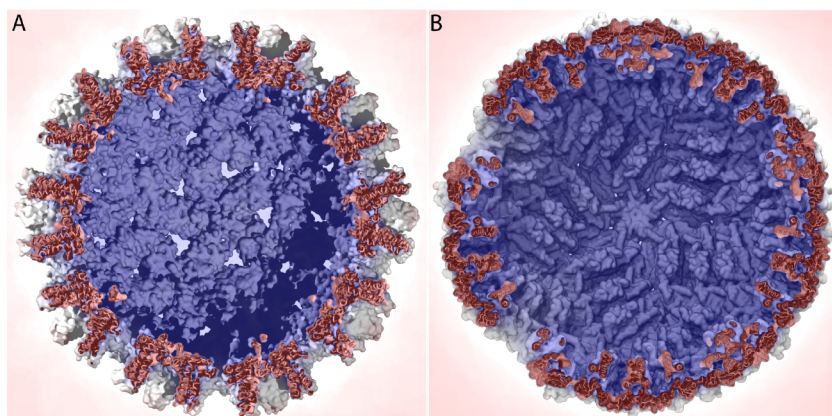


Figure 3. Visualization of outputs from *measure volinterior* with fuzzy-boundary detection. Two biomolecular container structures are shown in cross section, along with a volume slice from the continuous 3-D grid map produced by *measure volinterior* with fuzzy-boundary detection (e.g., *vol1*). (A) The pores of the HBV capsid²² must be masked with bulky QuickSurf parameters for accurate analysis with the fixed-boundary implementation of *measure volinterior*. The fuzzy-boundary detection method accurately identifies the capsid interior versus exterior despite the presence of pores. (B) Even with entire subunits removed from the ZIKV E/M protein shell,²³ the fuzzy-boundary detection method is able to delineate its interior versus exterior based on the extent to which these regions are occluded by the molecular surface.

Code Snippet 5 Assigning atom selection macros for use with *measure volinterior* fuzzy-boundary detection.

```
# Probability thresholds and intervals deemed appropriate for hypothetical System X
atomelect macro volinterior "vol1 > 0.95" ; # P above 90%, container interior for System X
atomelect macro volinterior "vol1 < 0.95" ; # P below 90%, container exterior for System X
atomelect macro voloccluded "vol1 > 0.74" ; # P above 75%, highly occluded region
atomelect macro volunoccluded "vol1 < 0.26" ; # P below 25%, relatively unoccluded region
atomelect macro volisdrange "vol1 > 0.40 and vol1 < 0.60" ; # P between 40-60%
```

Code Snippet 6 Measuring enclosed water density using *measure volinterior*.

```
set Nvoxels [measure volinterior $sel -res [expr $RadiusScale / 0.2]
             -isovalue $Isovalue -spacing $GridSpacing -nrays $n -mol $mol]
set WatersIn [atomelect $mol "water and oxygen and volinterior"]
set VoxelsIn [lindex $Nvoxels 2]
set Mw 18.0154
set V [expr $VoxelsIn * ($GridSpacing**3)]
set N [$WatersIn num]
set DensityIn [expr $Nw / (0.6022 * ($V/$N))]
# Note that density of TIP3P at 300 K and 1 atm is 1.002 g/cm**3
```

RESULTS AND DISCUSSION

Analysis of Biomolecular Containers. The fast and accurate detection of a biomolecular container's interior versus exterior by *measure volinterior* extends the ability of VMD's atom selection feature to collect the numbers and identities of particles that fall within these regions. The capability to efficiently track particle counts and locations over a series of trajectory frames enables a number of relevant container analyses, such as those presented below, even for multimillion-atom container systems.

Particle Densities. Biomolecular containers must regulate the density of particles, particularly water molecules, that they enclose in order to maintain correct solvent density. If the container is relatively impermeable, internal density that is too high or too low can lead to rupture or implosion, respectively. Moreover, tracking of enclosed particle densities represents a critical postprocessing assessment of any biomolecular container trajectory, which is necessary to ensure simulation equilibration and validity. Water density can be readily calculated using the relationship²⁶ given by

$$\rho = \frac{M_w}{0.6022 \times \left(\frac{V}{N}\right)} \quad (6)$$

where M_w is the molecular weight of the employed water model (e.g., 18.0154 g mol⁻¹ for TIP3P), V is the volume of the region of interest (i.e., container interior or exterior, in Å³, as shown for the HIV-1 capsid in Figure S4), and N is the number of particles located in the given region. A Tcl script demonstrating the measurement of water density within a biomolecular container using *measure volinterior* is presented in Code Snippet 6. Using this approach, the density of water both inside and outside of the HIV-1 capsid was found to be 1.000 ± 0.002 g cm⁻³, having equilibrated to the expected density for TIP3P after 100 ns of simulation.²⁷

Small-Molecule Transport. Biomolecular containers often play a role in regulating the biochemical processes they compartmentalize by controlling the passage of small molecules, ions, protons, or solvent.^{28,29} Many such containers act as semipermeable barriers, allowing the passage of particular species, while selectively filtering others. The transport properties of containers, or the rates at which they exchange molecular species across their surfaces, can be readily calculated by tracking the particles found inside versus outside over time. Notably, such exchange rates can be obtained using only MD simulations, which provide information on the container's surrounding solvent environment at the resolution of individual particles. For example, rates of solvent exchange have been recently reported for the HIV-1 and HBV capsids based on microsecond MD simulations, revealing that HIV-1 preferentially transports chloride, while HBV preferentially transports sodium.^{13,14}

A Tcl script demonstrating an approach to calculate solvent exchange rates using *measure volinterior* is presented in Code Snippet 7 (see the Supporting Information). Exchange is measured based on the identities of solvent species present on one side of the container at a reference frame t_0 and then found on the opposite side of the container at a subsequent frame $t > t_0$.^{13,14,30} The cumulative numbers of solvent species transported inward/outward per simulation time segment are exported to plotting software, and the slope of the linear fit to the inward/outward curves gives the respective rates. Examples of such fitting plots are available in the literature.^{13,14} Equivalence between inward and outward exchange rates indicates that the container system has reached equilibrium with its solvent environment. Importantly, the analysis aims to track transport of molecular species from bulk solvent inside the container to bulk solvent outside the container and vice versa. The accuracy of the calculation may be increased by

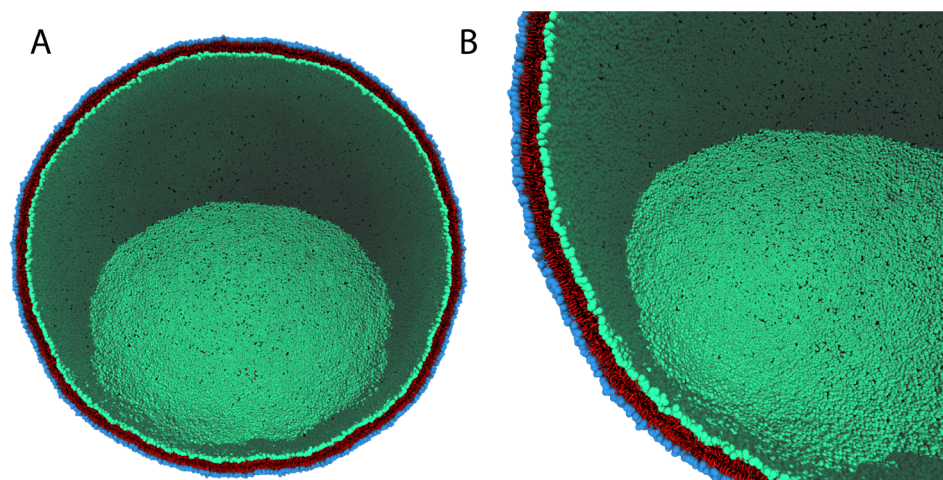


Figure 4. Detection of inner and outer leaflets of a lipid bilayer. A coarse-grained model of the HIV-1 envelope membrane is shown in cross section in (A) full and (B) quarter views. The parameters listed in Table 1 produce a molecular surface that corresponds to the structure of the lipid tail domains, and *measure volinterior* classifies the associated space as boundary region, colored red. Lipids located on the container interior, identified as members of the inner leaflet, have their head groups colored green; lipids located on the container exterior, identified as members of the outer leaflet, have their head groups colored blue. Leaflet classification is highly accurate, down to the level of individual particles.

using a slightly bulkier molecular surface for *measure volinterior*, which expands the boundary region between container interior and exterior and, thus, minimizes detection of recrossing events close to the structure, such as those that might arise when a species tends to localize within a container pore instead of exchanging directly between bulk solvent regions.

Transbilayer Lipid Diffusion. The transbilayer movement of lipids across membrane-bound containers, commonly referred to as lipid flip-flop, is an essential biological process.^{1,2} This exchange of lipids between the inner/outer leaflets can occur via free diffusion or protein-catalyzed translocation.³ Flip-flop is an important mechanism for membrane equilibration and enables the regulation of differing lipid compositions across the leaflets of a bilayer, a container property essential to the function of many organelles, including the plasma membrane.³¹ Rates of lipid flip-flop can be readily obtained by tracking lipids of a given species found in the inner and outer leaflets over time. Various experimental approaches have been developed to characterize protein-free lipid translocation,³² and transbilayer lipid diffusion can also be observed during MD simulations.

A Tcl script demonstrating the detection of the inner and outer leaflets of a membrane using *measure volinterior*, as well as an approach to track transbilayer diffusion, is presented in Code Snippet 8 (see the Supporting Information). Accurate detection of the inner and outer leaflets requires the parametrization of a molecular surface for *measure volinterior* that corresponds to the structure of the lipid tail domains sandwiched within the membrane bilayer (Figure 4). Given such a surface, the lipid head groups can be clearly identified as members of the inner and outer leaflets based on their locations inside and outside of the container core, respectively. Exchange between leaflets is measured based on the identities of lipid species present in one leaflet of the bilayer at a reference frame t_0 and then found in the opposite leaflet at a subsequent frame $t > t_0$. Rates of flip-flop observed on the simulation time scale can be determined based on plotting the cumulative numbers of lipid species transported to the inner/outer leaflet per simulation time segment and taking the slope

of the linear fit to the respective curves, similar to the analysis of solvent exchange rates discussed above.

Protein Cavities. Cavities embedded within protein structures are often key structural features that contribute to their biomolecular function. These regions may serve as binding sites for native ligands, form the openings of transport channels, or represent potential druggable sites. The identification and characterization of protein cavities is important for the study of fundamental biophysical properties like protein stability, elucidation of the molecular mechanisms underlying cavity-related protein function, and even the design of cavity-targeting therapeutics.^{33–36} Numerous computational approaches have been developed to address needs of the latter,³⁷ most aiming to perform some combination of cavity detection and visualization; analysis of cavity topology, volume, flexibility, and/or accessibility; prediction of cavity druggability scores; and/or selection of cavity conformers to generate ensembles for use in high-throughput computational screening of drug compounds. Popular software tools for the examination of cavities include POVME,^{38–40} fpocket,^{41,42} MDpocket,⁴³ TRAPP,^{44,45} Epock,⁴⁶ VolSite,⁴⁷ and trj_cavity.⁴⁸

The *measure volinterior* method provides a novel approach for detecting and measuring the spatial properties of protein cavities. Given that cavities are conceptualized essentially as containers, having interior and exterior regions, *measure volinterior* lends itself readily to the task. When protein cavities are buried and represent closed containers, analysis with the fixed-boundary *measure volinterior* implementation is sufficient to enable their characterization. For example, Figure 5A shows the results of applying *measure volinterior* to four structural variants of *Staphylococcal* nuclease (wild type, I92A, V66A, and L125A); the revealed cavity regions correspond closely to those previously identified for the system.⁴⁹ Importantly, the sensitivity of *measure volinterior* to molecular surface detail accurately captures variations in cavity size, shape, and intramolecular location arising from mutations, a biological phenomenon of known significance in drug development.^{50–52}

To enable visualization of the internal volumes of cavities identified by *measure volinterior*, such as shown in Figure 5A, the *Solvate* plugin in VMD can be used to generate a water box

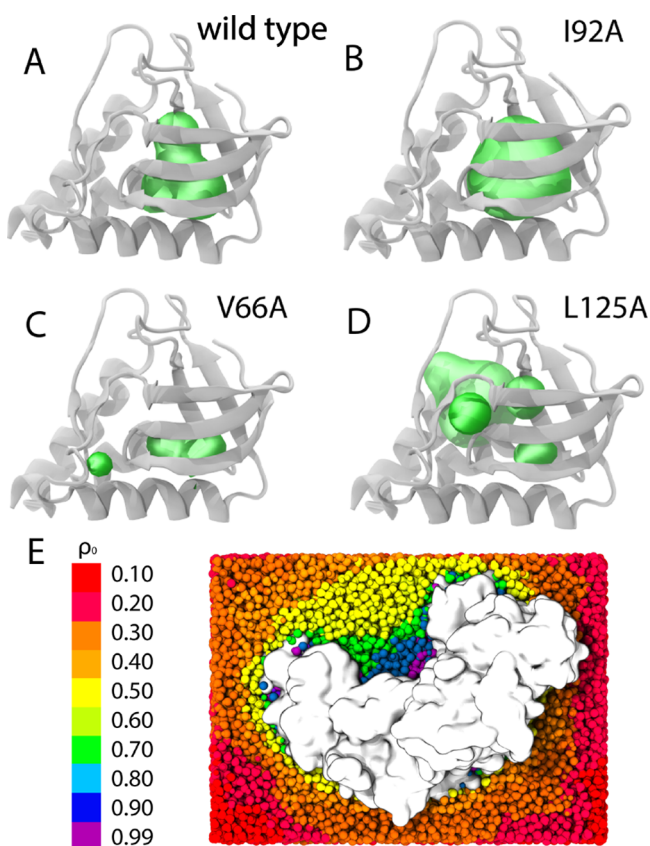


Figure 5. Detection of protein cavities. Four structural mutants of staphylococcal nuclease⁵⁵ are shown, with protein cavities identified using the fixed-boundary implementation of *measure volinterior* highlighted as green surfaces. (A) wild type, (B) I92A, (C) V66A, and (D) L125A. Cavities are detected as the interior of contained spaces buried within the proteins. Point mutations in the system lead to differences in cavity shape, volume, and intramolecular location. Parameters of *RadiusScale* = 0.9, *Isovalue* = 0.5, *Grid Spacing* = 1.3, and *N_{rays}* = 12 were used for *measure volinterior*. (E) The N-terminal domain of the glutamate receptor GluA1⁵⁴ is shown, immersed in solvent. Solvent molecules are colored according to the output of *measure volinterior* with fuzzy-boundary detection enabled. Probability threshold values, ρ_0 , correspond to the extent to which regions are occluded by the molecular surface and can be used to define the boundary of the cavity.

that overlays with the protein structure; by loading the 3-D grid output by *measure volinterior* into the water box, atom selections applied to water molecules that reference interior space (i.e., *volinterior*) appear as 3-D, space-filling representations within the cavity regions. The volume of a protein cavity can be calculated in the same way that the internal volume of a container is calculated in Code Snippet 6.

When protein cavities are solvent exposed and represent open containers, *measure volinterior*'s algorithm for fuzzy-boundary detection can be enabled. This approach allows the region that should be considered internal to the cavity to be defined based on the extent to which it is occluded by the molecular surface. In this way, fuzzy-boundary detection provides a novel strategy for handling the long-standing boundary definition problem for open binding pockets.⁵⁵ For example, Figure 5B shows the N-terminal domain of the glutamate receptor GluA1,⁵⁴ which exhibits a distinct central channel. The fuzzy-boundary detection method recognizes the channel as a cavity region, where voxels are highly occluded.

The boundary of the cavity, and thus its interior, can be established by selecting probability values above a threshold from the 3-D grid output by *measure volinterior*.

Trajectory Analysis with *measure volinterior*. The efficiency of *measure volinterior* enables analysis over entire simulation trajectories, with improved performance over other state-of-the-art software for container characterization. Benchmarks comparing volume classification results and run times for *measure volinterior*, Epock,⁴⁶ and *trj_cavity*⁴⁸ are presented in Figure S5 and Table S2, respectively. For a trajectory describing the Ebola virus ribonucleoprotein (RNP), including protein, RNA, solvent, and ions (4.8 million atoms) over 50 ns, *measure volinterior* processes each frame more rapidly, and without an *a priori* knowledge of the RNP structure required by the other methods. Importantly, fuzzy-boundary detection automatically identifies boundaries for the RNP cavity that can be tuned based on the selected probability threshold.

Overall, trajectory analysis with *measure volinterior*, particularly with fuzzy-boundary detection enabled, provides an extremely high-resolution characterization of container architecture at reduced computational expense. Time series plots for volume classification of the Ebola virus RNP, as well as a trajectory describing the HBV capsid, including protein, solvent, and ions (5.8 million atoms) over 1100 ns¹⁴ are relatively smooth, yet account for fine topological detail, yielding simultaneously greater accuracy and less noise compared to naïve geometric approaches to measure internal volume of the structures (Figure S3A, C). Volume slices of *measure volinterior*'s output 3-D grid show regions classified as the container interior conforming tightly to the molecular surface (Figure S3B, D). Notably, close tracking of molecular surface detail by *measure volinterior* extends the accuracy of the volume calculation for the HBV capsid beyond that obtained using even a sophisticated geometric approach,¹⁴ which captures equivalent trends of particle fluctuation (Figure S6A). Fuzzy-boundary detection automatically identifies boundaries for the capsid across all trajectory frames despite its large pores (Figure S6B).

High-Performance Container Analysis. In cases where container trajectories encompass multimillion-atom systems over extended simulation time scales, *measure volinterior* is well suited to take advantage of parallel, distributed-memory architectures through VMD's multiprocessor, GPU-enabled back-end, which can otherwise be bottlenecked by an I/O bandwidth in single-node cases. Code snippet 9 provides a Tcl script demonstrating how to perform container analysis on ORNL's Titan and Summit and NCSA's Blue Waters supercomputers using their high-performance Lustre file systems and VMD compiled with MPI (see the Supporting Information). With each trajectory frame striped across several object storage targets (OSTs), compute nodes benefit from parallel access, greatly decreasing the burden that I/O places on overall analysis execution time.⁵⁶ A schematic illustration of this process is shown in Figure 6.

As an example, the online storage available to users on NCSA's Blue Waters supercomputer is comprised of three Lustre storage systems with separate mount points (*/scratch*, */projects*, and */home*) that are available from both login and compute nodes. These volumes appear as standard disk partitions in the shell, granting users the capability to interface with the storage as part of a seemingly standard directory tree, avoiding the operational complexity of the underlying architecture. Once a target directory is created for the input

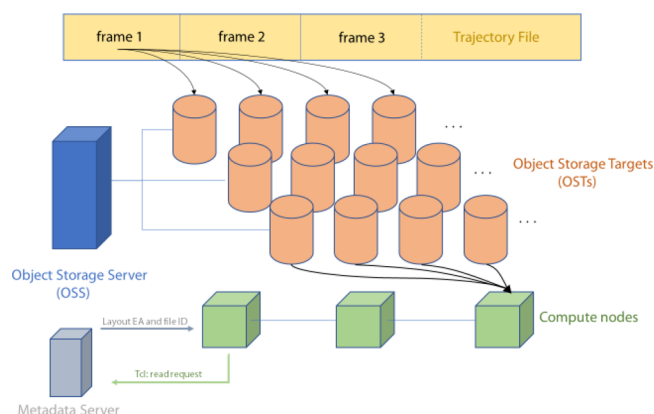


Figure 6. Trajectory frame analysis using the Lustre file system. Each frame in the trajectory file is striped onto separate object storage targets (OSTs). The amount of stripes is dependent on the size of the frame and the available memory in each OST. The object storage server (OSS) interfaces each OST with instructions from the client. Multiple instances of the client are spawned by each compute node once the job is submitted, and each compute node then processes a series of frames using parallel I/O to ameliorate bottlenecks. Instructions are organized through the metadata storage server (MDS), which returns memory addresses and extended attributes in response to high-level read requests sent through Tcl from clients.

trajectory, it can be striped to a specified count through the command-line.

When running an analysis using a similar framework to the aforementioned, the number of nodes is passed as a user-specified argument to *mpirun* during job submission and is input along with the given trajectory's frame count to block decompose a set of frames to each specific MPI rank. Each rank runs its own client instance and owns a unique index among all participating nodes. The instance running on node 0 ensures, via a conditional, that rank 0 is responsible for all printing. Once this information is primed for a given node, *parallel barrier* statements instruct each compute node to wait for all of its peers before *measure volinterior* analysis commences.

Using this strategy to perform container analysis on NCSA's Blue Waters supercomputer, the trajectory for the largest all-atom simulation ever published—an intact HIV-1 capsid in explicit solvent, encompassing 64 million atoms simulated over 1.2 μ s and boasting a data storage footprint of 70 TB¹³—can be processed in under 10 min of wall clock time. HIV-1 capsid trajectory analysis employed a stripe count of 120 and a parallel load of 12 frames per compute node.

CONCLUSION

The *measure volinterior* command implemented in VMD 1.9.4 represents a marriage between VMD's established molecular visualization and high-performance analysis features to realize a brand new capability: 3-D ray casting allows rapid classification of the space surrounding biomolecular containers regardless of container shape and constituent particle count, thereby enabling examination of containers on a scale that was previously unfeasible. Biomolecular containers are essential biological machines, whose study is critical to understanding the living cell. Containers such as enveloped virions,^{5,6} organelles,^{4,57} and potentially even minimal cells will no doubt be among the major systems investigated with MD simulations on upcoming exascale supercomputing platforms,

which will enable calculations encompassing billions of particles. Beyond enabling new scientific discoveries, large-scale MD simulations will continue to drive the development of novel technologies to overcome the challenges of working with big data sets.^{13,58} High-performance analysis software that can efficiently characterize the properties of biomolecular containers, particularly those that support tracking of solvent molecules over many trajectory frames, will drive the revelation of important discoveries arising from container simulations. With the new *measure volinterior* capability, VMD will enable researchers to routinely employ sophisticated biomolecular container analysis in the next supercomputing era.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.9b00324.

Information as mentioned in the text. (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: jperilla@udel.edu.

ORCID

Jodi A. Hadden-Perilla: 0000-0003-4685-8291

Juan R. Perilla: 0000-0003-1171-6816

Author Contributions

J.R.P. and J.E.S. designed the research. A.J.B., J.A.H.-P., J.E.S., and J.R.P. performed molecular modeling. A.J.B., J.A.H.-P., J.E.S., and J.R.P. wrote the *measure volinterior* code. A.J.B., J.A.H.-P., J.E.S., and J.R.P. validated results from the accelerated code. A.J.B., J.A.H.-P., J.E.S., and J.R.P. wrote the manuscript.

Notes

The authors declare no competing financial interest.

[§]A.J.B. and J.A.H.-P. are listed alphabetically and contributed equally to this work.

ACKNOWLEDGMENTS

We acknowledge funding by the National Institutes of Health Grants P50GM082251, P30GM110758-04, and P41GM104601 and the University of Delaware. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (Awards OCI-0725070 and ACI-1238993) and the state of Illinois. This work also used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation Grant ACI-1548562. An award of computer time was also provided by the Innovative and Novel Computational Impact on Theory and Experiment program (INCITE Award BIO024). This research also used resources from the Oak Ridge Leadership Computing Facility (OLCF) at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725. We acknowledge a Director's Discretionary award on the Summit supercomputer from the OLCF. Early VMD software development access to the Summit supercomputer was provided as part of the Center for Accelerated Application Readiness project at the OLCF.

REFERENCES

- (1) Contreras, F. X.; Sánchez-Magraner, L.; Alonso, A.; Goñi, F. M. Transbilayer (Flip-Flop) Lipid Motion and Lipid Scrambling in Membranes. *FEBS Lett.* **2010**, *584* (9), 1779–1786.
- (2) Daleke, D. L. Regulation of Transbilayer Plasma Membrane Phospholipid Asymmetry. *J. Lipid Res.* **2003**, *44* (2), 233–242.
- (3) Fattal, E.; Nir, S.; Parente, R. A.; Szoka, F. C., Jr Pore-Forming Peptides Induce Rapid Phospholipid Flip-Flop in Membranes. *Biochemistry* **1994**, *33* (21), 6721–6731.
- (4) Perilla, J. R.; Goh, B. C.; Cassidy, C. K.; Liu, B.; Bernardi, R. C.; Rudack, T.; Yu, H.; Wu, Z.; Schulten, K. Molecular Dynamics Simulations of Large Macromolecular Complexes. *Curr. Opin. Struct. Biol.* **2015**, *31*, 64–74.
- (5) Hadden, J. A.; Perilla, J. R. All-Atom Simulations of Viruses. *Curr. Opin. Virol.* **2018**, *31*, 82–91.
- (6) Reddy, T.; Sansom, M. S. Computational Virology: From the Inside Out. *Biochim. Biophys. Acta, Biomembr.* **2016**, *1858*, 1610–1618.
- (7) Marrink, S. J.; Mark, A. E. The Mechanism of Vesicle Fusion As Revealed By Molecular Dynamics Simulations. *J. Am. Chem. Soc.* **2003**, *125*, 11144–11145.
- (8) Marrink, S. J.; Mark, A. E. Molecular Dynamics Simulation of the Formation, Structure, and Dynamics of Small Phospholipid Vesicles. *J. Am. Chem. Soc.* **2003**, *125*, 15233–15242.
- (9) Knecht, V.; Marrink, S. J. Molecular Dynamics Simulations of Lipid Vesicle Fusion in Atomic Detail. *Biophys. J.* **2007**, *92*, 4254–4261.
- (10) Wu, S.; Guo, H. Simulation Study of Protein-Mediated Vesicle Fusion. *J. Phys. Chem. B* **2009**, *113*, 589–591.
- (11) Hofsäβ, C.; Lindahl, E.; Edholm, O. Molecular Dynamics Simulations of Phospholipid Bilayers With Cholesterol. *Biophys. J.* **2003**, *84* (4), 2192–2206.
- (12) Braun, A. R.; Sachs, J. N. Determining Structural and Mechanical Properties From Molecular Dynamics Simulations of Lipid Vesicles. *J. Chem. Theory Comput.* **2014**, *10* (9), 4160–4168.
- (13) Perilla, J. R.; Schulten, K. Physical Properties of the HIV-1 Capsid From All-Atom Molecular Dynamics Simulations. *Nat. Commun.* **2017**, *8*, 15959.
- (14) Hadden, J. A.; Perilla, J. R.; Schlicksup, C. J.; Venkatakrisnan, B.; Zlotnick, A.; Schulten, K. All-Atom Molecular Dynamics of the HBV Capsid Reveals Insights into Biological Function and Cryo-EM Resolution Limits. *eLife* **2018**, *7*, e32478.
- (15) Humphrey, W.; Dalke, A.; Schulten, K. VMD – Visual Molecular Dynamics. *J. Mol. Graphics* **1996**, *14* (1), 33–38.
- (16) Phillips, M.; Georgiev, I.; Dehof, A. K.; Nickels, S.; Marsalek, L.; Lenhof, H. P.; Hildebrandt, A.; Slusallek, P. Measuring Properties of Molecular Surfaces Using Ray Casting. In *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*; IEEE, 2010; pp 1–7.
- (17) Amanatides, J.; Woo, A. A Fast Voxel Traversal Algorithm for Ray Tracing. *Eurographics* **1987**, *87*, 3–10.
- (18) Bridson, R. Fast Poisson Disk Sampling in Arbitrary Dimensions. In: *SIGGRAPH sketches* **2007**, 22.
- (19) Krone, M.; Stone, J. E.; Ertl, T.; Schulten, K. Fast Visualization of Gaussian Density Surfaces for Molecular Dynamics and Particle System Trajectories. *EuroVis - Short Papers* **2012**, *2012*, 67–71.
- (20) Stone, J. E., Vandivort, K. L.; Schulten, K. GPU-Accelerated Molecular Visualization on Petascale Supercomputing Platforms. In *Proceedings of the 8th International Workshop on Ultrascale Visualization*; UltraVis '13, ACM: New York, 2013; pp 6:1–6:8.
- (21) Larson, S. B.; Day, J.; Greenwood, A.; McPherson, A. Refined Structure of Satellite Tobacco Mosaic Virus At 1.8 Å Resolution. *J. Mol. Biol.* **1998**, *277*, 37–59.
- (22) Bourne, C. R.; Finn, M. G.; Zlotnick, A. Global Structural Changes in Hepatitis B Capsids Induced by the Assembly Effector HAP1. *Journal of Virology* **2006**, *80*, 11055–11061.
- (23) Sirohi, D.; Chen, Z.; Sun, L.; Klose, T.; Pierson, T. C.; Rossmann, M. G.; Kuhn, R. J. The 3.8 Å Resolution Cryo-EM Structure of Zika Virus. *Science* **2016**, *352* (6284), 467–470.
- (24) Stehle, T.; Gamblin, S. J.; Yan, Y.; Harrison, S. C. The Structure of Simian Virus 40 Refined At 3.1 Å Resolution. *Structure* **1996**, *4* (2), 165–182.
- (25) Zhao, G.; Perilla, J. R.; Yufenyuy, E. L.; Meng, X.; Chen, B.; Ning, J.; Ahn, J.; Gronenborn, A. M.; Schulten, K.; Aiken, C.; Zhang, P. Mature HIV-1 Capsid Structure By Cryo-Electron Microscopy and All-Atom Molecular Dynamics. *Nature* **2013**, *497*, 643–646.
- (26) Jorgensen, W. L.; Jenson, C. Temperature Dependence of TIP3P, SPC, and TIP4P Water From NPT Monte Carlo Simulations: Seeking Temperatures of Maximum Density. *J. Comput. Chem.* **1998**, *19* (10), 1179–1186.
- (27) Perilla, J. R.; Zhao, G.; Lu, M.; Ning, J.; Hou, G.; Byeon, I. J. L.; Gronenborn, A. M.; Polenova, T.; Zhang, P. CryoEM Structure Refinement By Integrating NMR Chemical Shifts With Molecular Dynamics Simulations. *J. Phys. Chem. B* **2017**, *121*, 3853–3863.
- (28) Fuller, D. N.; Rickgauer, J. P.; Jardine, P. J.; Grimes, S.; Anderson, D. L.; Smith, D. E. *Proc. Natl. Acad. Sci. U. S. A.* **2007**, *104* (27), 11245–11250.
- (29) Nap, R.; Božič, A.; Szleifer, I.; Podgornik, R. The Role of Solution Conditions in the Bacteriophage PP7 Capsid Charge Regulation. *Biophys. J.* **2014**, *107* (8), 1970–1979.
- (30) Andoh, Y.; Yoshii, N.; Yamada, A.; Fujimoto, K.; Kojima, H.; Mizutani, K.; Nakagawa, A.; Nomoto, A.; Okazaki, S. All-Atom Molecular Dynamics Calculation Study of Entire Poliovirus Empty Capsids in Solution. *J. Chem. Phys.* **2014**, *141*, 165101.
- (31) van Meer, G.; Voelker, D. R.; Feigenson, G. W. Membrane Lipids: Where They Are and How They Behave. *Nat. Rev. Mol. Cell Biol.* **2008**, *9*, 112–124.
- (32) Allhusen, J. S.; Conboy, J. C. The Ins and Outs of Lipid Flip-Flop. *Acc. Chem. Res.* **2017**, *50* (1), 58–65.
- (33) Eriksson, A. E.; Baase, W. A.; Zhang, X. J.; Heinz, D. W.; Blaber, M.; Baldwin, E. P.; Matthews, B. W. Response of a Protein Structure To Cavity-Creating Mutations and Its Relation To the Hydrophobic Effect. *Science* **1992**, *255* (5041), 178–183.
- (34) Ohmura, T.; Ueda, T.; Ootsuka, K.; Saito, M.; Imoto, T. Stabilization of Hen Egg White Lysozyme By a Cavity-Filling Mutation. *Protein Sci.* **2001**, *10* (2), 313–320.
- (35) Brunori, M.; Vallone, B.; Cutruzzola, F.; Travaglini-Allocatelli, C.; Berendzen, J.; Chu, K.; Sweet, R. M.; Schlichting, I. *Proc. Natl. Acad. Sci. U. S. A.* **2000**, *97*, 2058–2063.
- (36) Pace, C. N.; Shirley, B. A.; McNutt, M.; Gajiwala, K. Forces Contributing To the Conformational Stability of Proteins. *FASEB J.* **1996**, *10* (1), 75–83.
- (37) 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. In *Computer Graphics Forum*; Wiley Online Library, 2016; Vol. 35, pp 527–551.
- (38) Durrant, J. D.; de Oliveira, C. A. F.; McCammon, J. A. POVME: an Algorithm for Measuring Binding-Pocket Volumes. *J. Mol. Graphics Modell.* **2011**, *29* (5), 773–776.
- (39) Durrant, J. D.; Votapka, L.; Sørensen, J.; Amaro, R. E. POVME 2.0: an Enhanced Tool for Determining Pocket Shape and Volume Characteristics. *J. Chem. Theory Comput.* **2014**, *10* (11), 5047–5056.
- (40) Wagner, J. R.; Sørensen, J.; Hensley, N.; Wong, C.; Zhu, C.; Perison, T.; Amaro, R. E. POVME 3.0: Software for Mapping Binding Pocket Flexibility. *J. Chem. Theory Comput.* **2017**, *13* (9), 4584–4592.
- (41) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: an Open Source Platform for Ligand Pocket Detection. *BMC Bioinf.* **2009**, *10* (1), 168.
- (42) Schmidtke, P.; Le Guilloux, V.; Maupetit, J.; Tuffery, P. Fpocket: Online Tools for Protein Ensemble Pocket Detection and Tracking. *Nucleic Acids Res.* **2010**, *38*, W582–W589.
- (43) Schmidtke, P.; Bidon-Chanal, A.; Luque, F. J.; Barril, X. MDpocket: Open-Source Cavity Detection and Characterization on Molecular Dynamics Trajectories. *Bioinformatics* **2011**, *27* (23), 3276–3285.
- (44) Kokh, D. B.; Richter, S.; Henrich, S.; Czodrowski, P.; Rippmann, F.; Wade, R. C. TRAPP: A Tool for Analysis of Transient

Binding Pockets in Proteins. *J. Chem. Inf. Model.* **2013**, *53*, 1235–1252.

(45) Stank, A.; Kokh, D. B.; Horn, M.; Sizikova, E.; Neil, R.; Panecka, J.; Richter, S.; Wade, R. C. TRAPP Webserver: Predicting Protein Binding Site Flexibility and Detecting Transient Binding Pockets. *Nucleic Acids Res.* **2017**, *45* (W1), W325–W330.

(46) Laurent, B.; Chavent, M.; Cragolini, T.; Dahl, A. C. E.; Pasquali, S.; Derreumaux, P.; Sansom, M. S.; Baaden, M. Epock: Rapid Analysis of Protein Pocket Dynamics. *Bioinformatics* **2015**, *31* (9), 1478–1480.

(47) Desaphy, J.; Azdimousa, K.; Kellenberger, E.; Rognan, D. Comparison and Druggability Prediction of Protein–Ligand Binding Sites From Pharmacophore-Annotated Cavity Shapes. *J. Chem. Inf. Model.* **2012**, *52*, 2287–2299.

(48) Paramo, T.; East, A.; Garzon, D.; Ulmschneider, M. B.; Bond, P. J. Efficient Characterization of Protein Cavities Within Molecular Simulation Trajectories: Trj_cavity. *J. Chem. Theory Comput.* **2014**, *10* (5), 2151–2164.

(49) Roche, J.; Caro, J. A.; Norberto, D. R.; Barthe, P.; Roumestand, C.; Schlessman, J. L.; Garcia, A. E.; Garcia-Moreno, B.; Royera, C. A. Cavities determine the pressure unfolding of proteins. *Proc. Natl. Acad. Sci. U. S. A.* **2012**, *109*, 6945–6950.

(50) Maurer-Stroh, S.; Ma, J.; Lee, R. T. C.; Sirota, F. L.; Eisenhaber, F. Mapping the sequence mutations of the 2009 H1N1 influenza A virus neuraminidase relative to drug and antibody binding sites. *Biol. Direct* **2009**, *4*, 18.

(51) Tantillo, C.; Ding, J.; Jacobo-Molina, A.; Nanni, R. G.; Boyer, P. L.; Hughes, S. H.; Pauwels, R.; Andries, K.; Janssen, P. A.; Arnold, E. Locations of Anti-AIDS Drug Binding Sites and Resistance Mutations in the Three-Dimensional Structure of HIV-1 Reverse Transcriptase: Implications for Mechanisms of Drug Inhibition and Resistance. *J. Mol. Biol.* **1994**, *243* (3), 369–387.

(52) Yun, C. H.; Mengwasser, K. E.; Toms, A. V.; Woo, M. S.; Greulich, H.; Wong, K. K.; Meyerson, M.; Eck, M. J. The T790M Mutation in EGFR Kinase Causes Drug Resistance By Increasing the Affinity for ATP. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105* (6), 2070–2075.

(53) Castaneda, C. A.; Fitch, C. A.; Majumdar, A.; Khangulov, V.; Schlessman, J. L.; Garcia-Moreno, B. E. Molecular Determinants of the PKa Values of Asp and Glu Residues in Staphylococcal Nuclease. *Proteins: Struct., Funct., Genet.* **2009**, *77* (3), 570–588.

(54) Yao, G.; Zong, Y.; Gu, S.; Zhou, J.; Xu, H.; Mathews, I. I.; Jin, R. Crystal Structure of the Glutamate Receptor GluA1 N-Terminal Domain. *Biochem. J.* **2011**, *438* (2), 255–263.

(55) Coleman, R. G.; Sharp, K. A. Protein Pockets: Inventory, Shape, and Comparison. *J. Chem. Inf. Model.* **2010**, *50* (4), 589–603.

(56) Stone, J. E.; Isralewitz, B.; Schulten, K. Early Experiences Scaling VMD Molecular Visualization and Analysis Jobs on BlueWaters. In: *Extreme Scaling Workshop (XSW)* **2013**, *2013*, 43–50.

(57) Chavent, M.; Duncan, A. L.; Sansom, M. S. P. Molecular Dynamics Simulations of Membrane Proteins and Their Interactions: From Nanoscale to Mesoscale. *Curr. Opin. Struct. Biol.* **2016**, *40*, 8–16.

(58) Goh, B. C.; Hadden, J. A.; Bernardi, R. C.; Singharoy, A.; McGreevy, R.; Rudack, T.; Cassidy, C. K.; Schulten, K. Computational Methodologies for Real-Space Structural Refinement of Large Macromolecular Complexes. *Annu. Rev. Biophys.* **2016**, *45*, 253–278.