

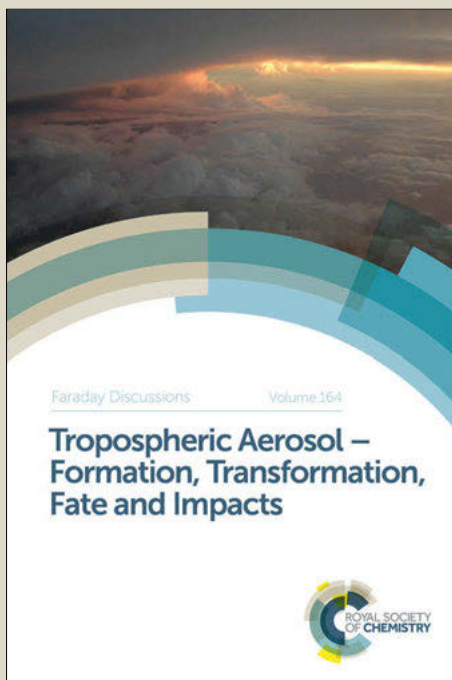
Faraday Discussions

Accepted Manuscript



This manuscript will be presented and discussed at a forthcoming Faraday Discussion meeting. All delegates can contribute to the discussion which will be included in the final volume.

Register now to attend! Full details of all upcoming meetings: <http://rsc.li/fd-upcoming-meetings>



This is an *Accepted Manuscript*, which has been through the Royal Society of Chemistry peer review process and has been accepted for publication.

Accepted Manuscripts are published online shortly after acceptance, before technical editing, formatting and proof reading. Using this free service, authors can make their results available to the community, in citable form, before we publish the edited article. We will replace this *Accepted Manuscript* with the edited and formatted *Advance Article* as soon as it is available.

You can find more information about *Accepted Manuscripts* in the [Information for Authors](#).

Please note that technical editing may introduce minor changes to the text and/or graphics, which may alter content. The journal's standard [Terms & Conditions](#) and the [Ethical guidelines](#) still apply. In no event shall the Royal Society of Chemistry be held responsible for any errors or omissions in this *Accepted Manuscript* or any consequences arising from the use of any information it contains.

GPU-Accelerated Analysis and Visualization of Large Structures Solved by Molecular Dynamics Flexible Fitting

John E. Stone,^{*a} Ryan McGreevy,^a Barry Isralewitz,^a and Klaus Schulten^{*b}

Received Xth XXXXXXXXXXXX 20XX, Accepted Xth XXXXXXXXXXXX 20XX

First published on the web Xth XXXXXXXXXXXX 200X

DOI: 10.1039/c000000x

Hybrid structure fitting methods combine data from cryo-electron microscopy and X-ray crystallography with molecular dynamics simulations for the determination of all-atom structures of large biomolecular complexes. Evaluating the quality-of-fit obtained from hybrid fitting is computationally demanding, particularly in the context of a multiplicity of structural conformations that must be evaluated. Existing tools for quality-of-fit analysis and visualization have previously targeted small structures and are too slow to be used interactively for large biomolecular complexes of particular interest today such as viruses or for long molecular dynamics trajectories as they arise in protein folding. We present new data-parallel and GPU-accelerated algorithms for rapid interactive computation of quality-of-fit metrics linking all-atom structures and molecular dynamics trajectories to experimentally-determined density maps obtained from cryo-electron microscopy or X-ray crystallography. We evaluate the performance and accuracy of the new quality-of-fit analysis algorithms vis-a-vis existing tools, examine algorithm performance on GPU-accelerated desktop workstations and supercomputers, and describe new visualization techniques for results of hybrid structure fitting methods.

1 Introduction

Molecular dynamics simulations provide researchers with a “computational microscope”, a powerful tool that provides dynamic views of cellular processes with atomic detail and nanosecond temporal resolution that can not be achieved through experimental methods alone. Petascale supercomputers extend the reach of the computational microscope to very large biomolecular systems of particular interest to public health, e.g., in the the case of viruses such as HIV¹.

Molecular dynamics simulations depend on the availability of all-atom molecular structures, but it can be extremely difficult to obtain all-atom atomic structures for large biomolecular complexes through traditional approaches. The most

^a Beckman Institute, University of Illinois, 405 N. Mathews Ave, Urbana, IL, USA.

^b Department of Physics, University of Illinois, 1110 W. Green, Urbana, IL, USA.

widely used method for acquiring structures of biomolecules is X-ray crystallography. However, crystallization of large biomolecules and macromolecular complexes can be challenging. Instead, cryo-electron microscopy single-particle reconstruction is becoming a central technique for structure determination of large biological complexes. Cryo-EM does not require the difficult crystallization step and allows the structure to be imaged in solution, more closely reproducing physiological conditions. Although cryo-EM yields sub-nanometer resolution data, sometimes approaching atomic resolution^{2–9}, crystallographic structures are generally still used for interpreting the cryo-EM data. This approach requires combining data from different imaging modalities using techniques known as hybrid fitting methods. Many hybrid fitting methods that combine X-ray crystallography structures and cryo-EM density for structure determination have been developed in recent years. Some of these methods use rigid-fragment fitting^{10,11}, while others such as Rosetta¹², DireX¹³, Gorgon⁵, and FRODA¹⁴ perform flexible fitting which allows conformational changes to better shape the structure to the density. Some approaches include the use of low-frequency normal modes¹⁵, deformable elastic networks¹³, and cross correlation¹⁶ or least-squares difference between experimental and simulated maps¹⁷ to drive the structure into the cryo-EM density. There are also fitting methods that use a Monte Carlo-based approach¹⁸, while others such as our own use molecular dynamics¹⁹.

Our method, Molecular Dynamics Flexible Fitting^{19,20} (MDFF), matches a crystal structure to a cryo-EM density potential energy function during an MD simulation. The additive modification of the potential energy function is defined on a 3-D grid and incorporated into an MD simulation using the `gridForces` feature of NAMD^{21,22}. Forces are computed from the added potential and applied to each atom depending on its position on the grid using an interpolation scheme. The computed forces push the atoms toward areas of higher density within the EM map. Restraints imposed during the simulation help preserve the secondary structure, stereochemical correctness²³, and symmetry²⁴ of the proteins. MDFF has proven to be successful, as evidenced by its many applications solving structural models for the ribosome^{25–33}, photosynthetic proteins^{34,35}, and the first all-atom structure of the HIV capsid¹.

An important step in any hybrid fitting method is the evaluation of the quality-of-fit of the final structure to the cryo-EM density. One of the most common scoring methods is the cross correlation coefficient between the experimental density map and a simulated density map. Other scoring functions exist, such as the Laplacian-filtered cross correlation or an envelope score which determines the amount of density filled with atoms, and use different approximations resulting in varying levels of accuracy³⁶. Approximations are useful because they enable the fast computation necessary for interactive visualization and analysis of fittings, particularly for large structures (> 1 million atoms) or long time-scale (microsecond range) simulations. Some scoring functions, including cross correlation, require the calculation of a simulated synthetic density map from the atomic structure which can then be compared to the experimental map. The traditional way of accomplishing this employs techniques created for X-ray crystallography which represent the density contribution of each atom as a spherical Gaussian function^{37,38}.

Below, we describe new data-parallel CPU and GPU algorithms for the rapid

computation of simulated density maps, density difference maps, cross correlation, and spatially localized cross correlation maps. The new algorithms serve both analytical and visualization uses and achieve performance levels that enable their effective use on large-size biomolecular complexes such as viruses, and for the detailed analysis of fitting results in the case of long MDFF trajectories. We illustrate the use of the new analysis features in several visualization scenarios and in the context of new MDFF quality-of-fit analysis features running on both desktop workstations and supercomputers.

2 Related Work

The original sequential CPU implementation of density simulation for MDFF was based on a method described earlier³⁹, which involves interpolating atomic weights onto a grid and convolving the grid with a Gaussian filter whose standard deviation is assumed to be half the target resolution. One problem with this implementation is that the spreading of atoms which occurs when interpolating the weights onto a grid is not accounted for when applying the Gaussian filter. The Situs⁴⁰ program `pdb2vol`⁴¹ takes a similar approach to creating synthetic densities, but it corrects for the lattice smoothing by subtracting the lattice projection mean-square deviation from the Gaussian kernel variance. Situs also allows the user to select kernel types other than Gaussian, such as triangular or hard sphere. For the Situs-generated density maps created for the comparisons reported in this paper, a Gaussian kernel with lattice smoothing correction and mass-weighted atoms was used (Situs version 2.7.1 obtained from <http://situs.biomachina.org/>). Chimera⁴² produces synthetic density maps with the `molmap` command by describing each atom as a 3-D Gaussian distribution of width proportional to the map resolution and amplitude proportional to the atomic number; the command is based on the `pdb2mrc` program of the EMAN package⁴³. All Chimera calculations reported in this paper employed the `measure correlation` command with default parameters, except for `aboveThreshold` set to false. The resulting correlation about mean values were used as the point of comparison (Chimera version 1.8.1 obtained from <https://www.cgl.ucsf.edu/chimera/>).

All of these methods are similar in that they use spherical Gaussian functions to represent the density contribution of each atom, a technique created for X-ray crystallography^{37,38}. One shortcoming of this approach is that it does not incorporate the blurring required to account for the attenuation of high resolution signal¹⁷. Approximations for the effect of truncation were derived for X-ray crystallography³⁷, but not for EM applications. One method for overcoming this limitation is to create the map from simulated densities of each atom calculated by a resolution-limited Fourier transform of the atomic scattering factor⁴⁴. This is the approach taken in DireX¹³ which also uses an *ad hoc* blurring method to account for the instrumental signal attenuation at high resolution. Another program, RSRef, implements a similar method which does not rely on any *ad hoc* assumptions¹⁷.

We compared densities produced by our new GPU-based algorithms and by Chimera against those generated from DireX for the ribose binding protein (1URP). The cross correlations for a 3 Å and 8 Å map were .938 and .966, respectively, for our method, and .900 and .965 for Chimera. These correlation values

	RHDV	1URP	1URP
Resolution (Å)	6.5	8	3
VMD-GPU-CUDA cc	0.998	0.999	0.989
VMD-CPU-SSE cc	0.996	0.997	0.991
Chimera cc	0.784	0.989	0.922
VMD-CPU-SEQ cc	0.997	0.999	0.950

Table 1 Comparison of cross correlation results for the new GPU and CPU algorithms in VMD, for Chimera, and for the original sequential VMD algorithm, on densities simulated using the Situs⁴⁰ program `pdb2vol`⁴¹. These tests show the agreement between the densities generated by each of the algorithms and account for any discrepancies in the comparison of subsequent cross correlations on experimental data.

are, as expected, lower than correlation values for densities created by Situs (Table 1), though the VMD and Chimera values are still within 5% of each other. These differences may be more important for fitting methods which, similar to `RSRef`, directly minimize the least-squares difference between the simulated and experimental density maps. Conversely, MDFF derives forces from the experimental map alone and, thus, uses the cross-correlation only in subsequent analysis steps. The more accurate density synthesis methods which utilize Fourier transforms require much more computation than the Gaussian approximations. Future work could experiment with hybrid approaches that borrow aspects from both methods to improve accuracy without sacrificing speed of computation. The goal of the method described here is fast computation suited to visualization and analysis, a use which is amenable to approximation.

3 Hybrid Fitting Process

In many hybrid methods, the initial structure must first be docked into the EM density before further fitting can proceed. This process is generally an expedient way of rigidly docking the entire structure, or several independent rigid pieces, into the density so that they are correctly oriented within the map. To achieve this with MDFF, rigid-body docking is performed using the Situs⁴⁰ program `colores`⁴⁵. Once the structure has been docked to the density, it is often useful at this stage to calculate the quality-of-fit. In particular, independent local correlations of each domain or secondary structure are useful in order to visualize which areas of the structure are initially poorly fit to the map. This quickly identifies areas that may be of greatest concern during the subsequent fitting simulation.

The next step in the fitting process is the determination of the finer-grained movements and any large scale deformations that are required to improve the fit to the reference map. For MDFF, this is the point where molecular dynamics simulations are used to flexibly guide the structure into the density. The VMD⁴⁶ plugin `mdff` is used to generate a potential energy function U_{EM} defined on a

3-D grid derived from the cryo-EM density map

$$U_{\text{EM}}(\mathbf{R}) = \sum_j w_j V_{\text{EM}}(\mathbf{r}_j), \quad (1)$$

where

$$V_{\text{EM}}(\mathbf{r}) = \begin{cases} \xi \left[1 - \frac{\Phi(\mathbf{r}) - \Phi_{\text{thr}}}{\Phi_{\text{max}} - \Phi_{\text{thr}}} \right] & \text{if } \Phi(\mathbf{r}) \geq \Phi_{\text{thr}}, \\ \xi & \text{if } \Phi(\mathbf{r}) < \Phi_{\text{thr}}. \end{cases} \quad (2)$$

Here $\Phi(\mathbf{r})$ is the Coulomb potential; it and its maximum value Φ_{max} are obtained from cryo-EM. In Eq. 2 a threshold Φ_{thr} is used to clamp all $\Phi(\mathbf{r})$ values which are lower than Φ_{thr} , effectively removing the solvent contribution from the map and creating a flat potential for those regions. This threshold Φ_{thr} is chosen to be at or above the value at which a histogram of the EM density contains a large peak corresponding to the solvent. The scaling factor ξ uniformly adjusts the strength of the influence of the cryo-EM map on the molecular system. In addition, $V_{\text{EM}}(\mathbf{r})$ has a weight w_j for each atom j present at position \mathbf{r}_j . Generally, w_j is set to the atomic mass which follows from the roughly linear correspondence between the mass of atoms and their density in a cryo-EM map.

Forces computed from the potential defined in Eq. 1 push the structure into regions of high density. This push is often divided into a multi-stage process where, in the various stages, parts of the atomic structure can be completely de-coupled from the potential, or have the magnitude of the force varied by changing the scaling factor ξ or weights w_j . In the case of the ribosome^{25–33}, protein and RNA are often each fit separately while the other is constrained. In such a multi-stage scheme, it is beneficial to calculate the quality-of-fit at each step separately for each component of the molecular complex, enabling the identification of problematic regions.

For many systems, once the entire structure has been fit, a single correlation value for the entire protein will be published, giving a coarse overall evaluation of the model. Because this value may be used for comparison across different methods or studies, an accurate, reproducible, and comparable quantity must be used. However, it is often useful to calculate the correlation at a finer decomposition (Fig. 1), for example per-residue. Such an analysis could resolve which parts of the structure are fitting well and which parts might require additional fitting or investigation. Additionally, it can be beneficial to determine how local correlations change, and hopefully improve, over the course of the fitting. A visualization of a fine-grained quality-of-fit over the entire simulation can provide the researcher with insight into the progression of the fitting process and the dynamics of each part of the structure. Computing the simulated densities for many different regions and their subsequent correlations for each step of the fitting can be extremely demanding computationally, especially for large structures or long fitting simulations. Practical application would require efficient algorithms permitting fast analysis turnaround.

The VMD TIMELINE trajectory analysis plugin⁴⁷ makes use of the new data-parallel CPU and GPU quality-of-fit algorithms to provide an interactive visualization of the entire fitting process. TIMELINE is designed to help identify and assess trajectory events by performing analysis calculations for each component of a molecular system, and for every frame of a simulation trajectory. The

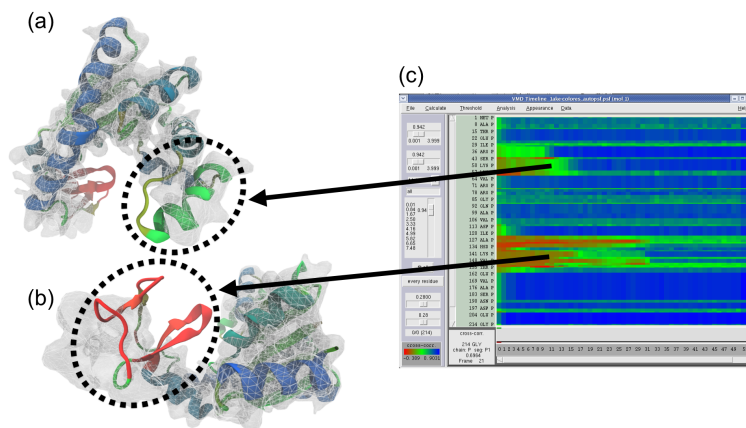


Fig. 1 Per-secondary-structure-component local cross correlation of adenylate kinase. MDFF was run on an initial conformation of adenylate kinase (PDB 1AKE) in order to fit it to a simulated density at 5 Å resolution from another conformation (PDB 4AKE). Panels (a) and (b) show a point during the simulation before the structure has been completely fit, colored by per-secondary-structure-component local cross correlation. Darker blue indicates the areas of best fit, while red indicates poorly fitting regions. The target density is drawn as a transparent gray isosurface with a complementary wireframe representation. Panel (a) shows a helix partly in the density; the helix is colored green to indicate a mediocre cross correlation indicative of a relatively close, but not perfect, fit to the density. Panel (b) shows a beta sheet almost completely outside of the reference density; the beta sheet is colored red to indicate a poor cross correlation. (c) VMD TIMELINE analysis of the cross correlation over the entire simulation. Note how regions of poor fit (red) turn green and blue as the fit improves during the course of the simulation.

heatmap-style 2-D matrix TIMELINE plot provides a “whole-trajectory” and a “whole-structure” view of the calculated property. Figure 1c shows an example plot: the time dimension is displayed horizontally, the structure-component dimension is displayed vertically, and the property of interest – here, density map cross correlation score – is presented by colored rectangles. The TIMELINE plot is connected interactively to the VMD 3-D structure display. When the user “scrubs” the mouse cursor around features on the plot, TIMELINE highlights the associated 3-D molecular structures and displays their configurations and motions at the times of the corresponding events. A researcher can identify structure transitions, trends, and sequences in a trajectory more quickly by using a single zoomable, interactive 2-D TIMELINE plot than by examining a complex 3-D structure over many trajectory frames. When calculating density map cross correlations, different types of trajectory information can be obtained by dividing the system into analysis components of different sizes: fine division among individual residues or contiguous secondary structures (as in Figure 1c) allows detailed structural assessment, while coarser division to fractions of protein segments or even entire protein segments provides information about larger structural elements, particularly for larger systems.

If quality-of-fit analysis can be performed fast enough for interactive updates to keep pace with a running simulation, the correlations can be visualized by the user in real time. Such a real-time analysis can enable the user, along with interactive molecular dynamics (IMD)⁴⁸, to monitor fitting progress and manually manipulate parts of the structure. The user could move the structure in an attempt to improve the correlation while being provided immediate feedback about the resulting effects, with trajectory and local trend context provided by an updating TIMELINE view. This technique could make the fitting of difficult structures a much easier process while providing detailed analysis at every step. Additionally, this method could help fit small regions which normally take a long time to fit, often well after most of the structure has long since converged. Interactive MDFF has been performed in the past for studies on the ribosome²⁹, but real-time quality-of-fit analysis could provide for better fits and easier simulations.

4 Methods for Evaluating Quality of Fit

In MDFF, the simulated density is compared to the experimental map using Pearson's correlation coefficient, given by

$$\rho_{SE} = \frac{\langle (S - \langle S \rangle)(E - \langle E \rangle) \rangle}{\sigma_S \sigma_E}, \quad (3)$$

where $\langle S \rangle$ and $\langle E \rangle$ are the average voxel values of the simulated and experimental density maps, respectively, and σ_S and σ_E are their standard deviations⁴⁹. The result is a normalized ($\rho_{SE} \in [-1, 1]$) indicator of the quality-of-fit of a given structure in the experimental density. The cross correlation defined in Eq. 3 is traditionally used as a global measure that includes all voxels in the density maps, or as a local correlation¹¹ where only the volume surrounding the molecule is considered. The global correlation is affected by the size of the volume and includes contributions not only from the molecule, but also the solvent. Global correlations can report artificially high correlation values when considering large volumes that have been arbitrarily padded. Accordingly, local correlations tend to offer a more useful and unbiased characteristic. In MDFF, local correlations are calculated by specifying a threshold stated in units of standard deviations from the mean value of the simulated density. All voxels containing density values lower than this threshold are excluded from the cross correlation calculation. With higher density values naturally occurring closer to the structure, the density being used for the calculation is thus contained within the molecular envelope.

One impediment to the creation of a fast algorithm for computation of the cross correlation is the need for knowledge of the mean value of each map for both the covariance and standard deviations. A simple two-pass algorithm would first iterate through the maps and compute the mean, followed by a second iteration which would compute the covariance and standard deviation. In the interest of speed and efficiency, a single-pass algorithm is desirable, where only one iteration over the densities would be required. For a single-pass version of the standard deviation calculation, the equation can be rewritten

$$\sigma_S^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 = \frac{1}{N} \sum_{i=1}^N (x_i^2 - 2\bar{x}x_i + \bar{x}^2) = \frac{1}{N} \sum_{i=1}^N x_i^2 - \frac{2\bar{x}}{N} \sum_{i=1}^N x_i + \frac{N\bar{x}^2}{N} \quad (4)$$

$$= \left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right) - 2\bar{x}^2 + \bar{x}^2$$

or

$$\sigma_S^2 = \left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right) - \bar{x}^2. \quad (5)$$

This expression permits one to compute the sum and the sum of squares of the voxels in each map in a single pass, with the standard deviation being calculated afterwards. This technique can be extended similarly to create a single-pass formulation of covariance. Unfortunately, when the single-pass equations are implemented in floating point arithmetic, they are susceptible to floating point truncation and cancellation error, which can affect the accuracy of the results. Floating point error can be minimized by shifting the means of the reference and simulated maps to zero, and by ensuring that the simulated density maps closely approximate the standard deviation of a reference map, thereby reducing the magnitudes of accumulated squared quantities. Further improvements to floating point precision can be achieved by performing summations in a hierarchical fashion, by using double-precision arithmetic, or through the use of precision enhancement techniques such as Kahan's compensated summation, native-pair arithmetic, and others^{50–52}. In our tests, we have confirmed the presence of floating error, but even without implementing all of the amelioration techniques described above, we have concluded that the errors have a negligible effect on the final correlation results for the test cases we have run to date. We believe this to be the direct result of a well-conditioned input scenario due to the limited dynamic range of the density values contained within both experimental and simulated density maps.

5 Data-Parallel Cross-Correlation Algorithms

We have devised high-performance data-parallel algorithms for computing simulated density maps, and comparing them with reference density maps using multi-core CPUs with vector instructions and massively parallel GPUs. Each of the methods we have devised for evaluating the quality-of-fit between an all-atom structure and a reference experimental density map requires the computation of a simulated density map from the all-atom structure. Once computed, the simulated density map is then compared with the experimental reference map using voxel-by-voxel subtraction, Pearson correlation, sum of absolute differences, and similar methods. In all of the methods we have evaluated thus far, the generation of the simulated density map is by far the most computationally demanding algorithm step. Since the simulated density value at a given point in space can be computed independently from the values of all other points, the simulated density map computation is well-suited to data-parallel implementations using CPU vector instructions such as the Intel x86 SSE and AVX instructions, or the ARM NEON instructions. Similarly, the computation can take advantage of GPU-accelerated algorithms that employ tens of thousands of threads running on thousands of arithmetic units. Once the simulated density map has been computed, it can be compared with the experimental reference map using a variety of methods.

Single-Pass Cross-Correlation Parallel Decomposition

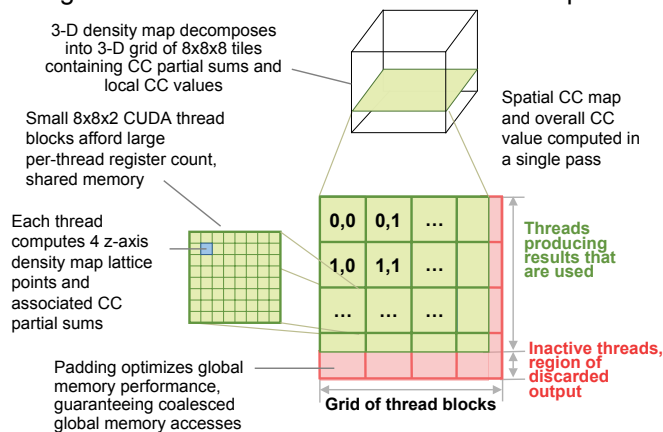


Fig. 2 Parallel decomposition for the single-pass GPU-accelerated cross correlation algorithm. The simulated density map is decomposed into $8 \times 8 \times 8$ tiles which are processed by individual CUDA thread blocks. Each thread block is $8 \times 8 \times 2$ threads in size. Each thread computes 4 consecutive densities along the z-axis, thereby enabling reuse of atom data in on-chip registers and increasing arithmetic throughput. Each CUDA thread block performs thresholding and computes partial sums for the overall cross correlation, as well as the local cross correlation value for its $8 \times 8 \times 8$ tile.

A key difference between different density map comparison techniques lies in their respective memory capacity requirements and, in the case of GPU implementations, the host-GPU memory transfer characteristics of each algorithm. It is a straightforward task to implement quality-of-fit algorithms using multi-pass approaches that compute densities, means, and covariances in distinct steps, but these approaches lose performance relative to implementations that combine all steps into a single-pass algorithm. By combining all algorithm steps into a single-pass approach, intermediate values can be stored in fast on-chip registers, shared memory, and caches, that are typically two orders of magnitude faster than accesses to off-chip DRAM. When possible, we employ single-pass cross correlation algorithms, where the simulated density values are computed only ephemerally in CPU or GPU on-chip registers without ever storing the computed density values to off-chip DRAM memory. This approach avoids performance limitations that arise from the roughly factor of 40 performance gap between peak GPU arithmetic throughput (e.g., 2-4.5 TFLOPS) and GPU DRAM memory bandwidth (e.g., 200-300 GB/sec).

We have previously developed fast algorithms for calculating and visualizing molecular orbitals present in quantum chemistry calculations⁵³, and for display of molecular surfaces, biomolecular complexes, and cellular structures⁵⁴⁻⁵⁶. These algorithms each require rapid evaluation of Gaussian radial basis functions, which is also necessary for computation of simulated density maps used in computing cross correlations. In the present work, the simulated density map is

computed from a linear combination of Gaussian radial basis functions, with a per-atom weighting factor based on atomic radii. The simulated density map can be computed in a single-pass, for single-coefficient atomic weighting factors, or in multiple passes when creating temporally averaged density maps, and to support the use of more sophisticated schemes that use radial basis functions with multiple Gaussian terms or other basis functions. The reference density is tested for exclusion, and the simulated densities are compared with an optional user-defined threshold criterium to determine if they should contribute to the cross correlation. Reference voxels assigned with the IEEE floating point special value NaN (not-a-number), are excluded from consideration. Similarly, simulated density values that fall below an optional user-defined density threshold are also excluded from contributing to the cross correlation. All voxels for which both the reference and simulated densities pass the exclusion and thresholding tests contribute then to the partial sums for the overall cross correlation. When all densities have been processed, a parallel reduction is performed to compute the final cross correlation from all partial sums that were previously computed.

The new single-pass cross correlation algorithm for GPUs is based on the NVIDIA CUDA programming toolkit⁵⁷, and is referred to hereafter as VMD-GPU-CUDA. The GPU cross correlation algorithm uses a parallel decomposition outlined in Fig. 2. Density contributions from neighboring atoms within the cutoff distance are summed into thread-local registers. The cutoff distance is chosen such that density contributions at or sufficiently near zero are discarded, thereby ensuring that the algorithm achieves linear time complexity. Each thread produces four density values by looping over consecutive densities along the z-axis. Each thread computes its own local contributions to the per-tile cross correlation partial sums, for its four densities according to the voxel exclusion and thresholding criteria described above. Each CUDA thread block performs a local intra-block parallel reduction among all threads, summing each of the thread-local partial sums into a single set of cross-correlation partial sums associated with that block's $8 \times 8 \times 8$ tile. Once the intra-block parallel reduction is complete, the first thread of the thread block computes a tile-local cross correlation, and stores both the tile-local cross correlation as well as the per-tile cross correlation partial sums to GPU global memory, in preparation for a final inter-block global parallel reduction that yields the overall cross correlation result. One important performance-critical detail about our implementation is that we employ special counters and atomic memory update instructions to allow a single GPU kernel invocation to perform the entire cross correlation calculation. Each thread block increments a global counter using an *atomic-add* machine instruction as it completes its tile-local work. All but the last thread block to complete simply exit once they have finished their tile-local work. The last thread block to complete its tile-local work performs the final inter-block parallel reduction of all partial sums and stores the overall cross correlation result to GPU global memory.

The new cross-correlation algorithm for multi-core CPUs employs CPU vector instructions and multithreading. The new CPU algorithm is hereafter referred to as VMD-CPU-SSE. The CPU algorithm decomposes the density map into planar slices which are dynamically assigned to a fixed pool of CPU worker threads by a dynamic load balancer built into VMD⁴⁷. Each CPU worker thread runs a loop that accumulates densities from each atom onto the region surrounding

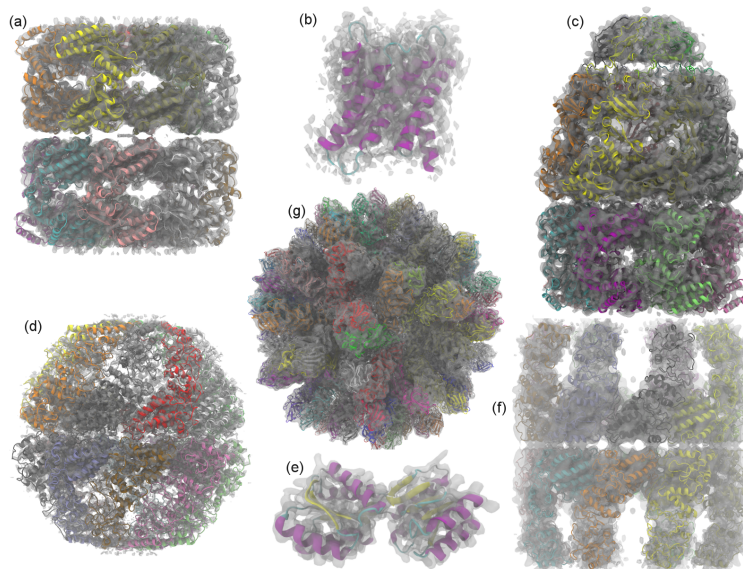


Fig. 3 All test systems fitted within their experimental density, or in the case of 1URP, a synthetic one. (a) GroEL PDB 3E76; (b) aquaporin-0 PDB 3M9I; (c) GroEL-GroES complex in the ATP-bound state (PDB 2C7D); (d) Mm-cpn in the closed state (PDB 3LOS); (e) D-ribose-binding protein PDB 1URP; (f) lidless Mm-cpn in the open state; PDB 3LOS was used as a template to construct a homology model using Modeler⁵⁸; (g) rabbit hemorrhagic disease virus capsid (RHDV) (PDB 3J1P).

the atom up to a defined cutoff limit where the Gaussian is assumed to have decayed to zero. Mainstream multi-core CPUs lack dedicated machine instructions for calculation of exponentials; as a result, application software must either use general purpose math library routines or incorporate custom functions for this purpose. Since the sign of the exponential function's input domain is always negative for the density map synthesis algorithm we have developed, we use a high-performance Taylor series approximation of the exponential function. Our specialized negative-domain exponential approximation is hand-written in assembly language intrinsics for Intel x86 SSE 4-way vector instructions, based on a similar function we originally developed for molecular orbital display⁵³. We have improved the performance of the new exponential approximation beyond what we had previously achieved by pre-multiplying input values with constants to convert to base- e , by eliminating conditional tests that check for input domain cases that cannot occur in our density map algorithm, and by guaranteeing multiple-of-four data alignment. The end result of these optimizations is that the multi-core CPU density map algorithm can be competitive with the performance obtained with GPUs, which have machine instructions for computing exponentials due to their heavy use in computer graphics lighting algorithms.

	RHDV	Mm-cpn open	GroEL	GroEL- GroES	Mm-cpn closed	Aquaporin
Resolution (Å)	6.5	8	4	7.7	4.3	3
# atoms	702K	61K	54K	59K	64K	1.6K
# voxels	111M	14M	8M	7M	7M	2.6M
VMD-GPU-CUDA cc	0.744	0.857	0.769	0.886	0.658	0.754
VMD-CPU-SSE cc	0.736	0.866	0.839	0.892	0.712	0.771
Chimera cc	0.687	0.837	0.727	0.867	0.635	0.739
VMD-CPU-SEQ cc	0.743	0.869	0.846	0.898	0.728	0.653
VMD-GPU-CUDA	0.458 s 34.6×	0.06 s 25.7×	0.034 s 36.8×	0.038 s 35.0×	0.028 s 57.5×	0.007 s 55.7×
VMD-CPU-SSE	0.779 s 20.3×	0.085 s 18.1×	0.159 s 7.9×	0.077 s 17.3×	0.179 s 9.0×	0.033 s 11.8×
Chimera	15.86 s 1.0×	1.54 s 1.0×	1.25 s 1.0×	1.33 s 1.0×	1.61 s 1.0×	0.39 s 1.0×
VMD-CPU-SEQ	62.89 s 0.25×	2.9 s 0.53×	1.57 s 0.79×	2.49 s 0.53×	2.13 s 0.75×	0.04 s 9.7×

Table 2 Comparison of cross correlation algorithm results, runtimes, and speedups for the new GPU and CPU algorithms in VMD, for Chimera, and for the original sequential VMD algorithm. All speedups are normalized using Chimera runtime as the baseline.

6 Test Cases and Performance Results

In order to test the new data-parallel GPU and CPU cross correlation algorithms, nine test cases were chosen from structures of varying size with density maps at resolutions between 3 Å and 8 Å (Fig. 3, Table 1, Table 2). Six of the structures are final results of MDFFF simulations with their corresponding experimental cryo-EM maps, while the remaining three are structures with simulated density maps using the Situs⁴⁰ program *pdb2vol*⁴¹. The largest structure is the rabbit hemorrhagic disease virus (RHDV) capsid solved with a 6.5 Å resolution density map⁵⁹. Structures of aquaporin, GroEL, GroEL-GroES complex in the ATP-bound state, Mm-cpn in the closed state, and lidless Mm-cpn in the open state, were taken from the results of a cryo-EM modeling challenge⁶⁰. The final structure, ribose binding protein (PDB: 1URP) was used to simulate density maps at 3 Å and 8 Å using Situs, as a means of comparing to density maps generated from another program (Table 1). The size of the virus capsid and the difficult-to-fit nature of the modeling challenge structures both pose problems to researchers using hybrid fitting methods. Our data-parallel GPU and multi-core CPU cross correlation methods were compared to the previous sequential algorithm in VMD, as well as to Chimera⁴², a program widely used by cryo-EM researchers.

The new VMD cross correlation algorithms provide a significant speedup over both the old sequential CPU code in VMD as well as Chimera. The algorithms provide correlation results that agree with Chimera to within 5% for most cases. In the case where the two have the greatest divergence (7.92% dif-

ference), the virus capsid, our method results in a .998 cross correlation with a simulated map produced by Situs, while Chimera reports a correlation of .7842 (Table 1). Because of this discrepancy on the simulated data, we believe that the correlation result for the large capsid reported by Chimera might be flawed, or that it is otherwise not necessarily directly comparable in that specific case. One point of possible error is the difference between VMD and Chimera in handling PDB files with over 99,999 atoms. The original PDB specification allows for the use of only 5 columns for the atom index, restricting the maximum officially supported number of atoms to 99,999. To overcome this limitation, VMD uses alpha-numeric atom indices beyond 99,999 when writing out a PDB file, while Chimera uses additional columns. Because VMD can read these large structure PDB files written in Chimera format but not vice versa, a Chimera format PDB was used in all cases (including RHDV) where it was required, for both the VMD and Chimera calculations. Chimera format PDB files were created by first writing the structure coordinates into multiple PDB files of 99,999 atoms or less. The PDB files were then read into Chimera, followed by using the "copy/combine" feature to combine every PDB into a single structure which could be saved to a single PDB file. An additional overlooked complication regarding the difference in PDB file formats could be the cause of the discrepancy in correlation values obtained for RHDV.

The new feature for calculating spatial cross correlation maps provides a rapid means of visualizing 3-D correlations and resolving areas of poor fit, as seen in Fig. 4. These spatial correlation maps are especially useful for large complexes, for example virus capsids, because only a single density calculation is required to obtain localized information. By using the VolumeSlice representation in VMD to view 3-D textured slice planes of the correlation map in the x -, y -, or z - axis, the map can be quickly scanned for regions of low correlation. Structure components located in regions of poor correlation can be identified by selecting them or coloring them based on local cross correlation values (Fig. 5).

All single-node workstation class tests were run on a workstation containing two Intel Xeon E5-2687W processors each consisting of 8 physical processor cores and 16 hardware threads, for a total of up to 32 threads of execution in hardware. The workstation was configured with 256 GB of memory, and an NVIDIA Quadro K6000 GPU with 12 GB of on-board memory. All software was run on CentOS Linux version 5.9, and NVIDIA driver version 331.22. VMD was compiled using Intel C/C++ version 12.0.2, and CUDA toolkit version 4.0. Tests involving Situs used Situs version 2.7.1, as obtained from the Situs web site. Chimera tests used version 1.8.1 as obtained from the Chimera web site.

The accelerated cross correlation method makes feasible both detailed and global trajectory assessments of large molecular systems. Figure 6 shows an example of the diagnostic insight provided by a TIMELINE view of the initial part of an MDFF simulation of RHDV capsid: 100 frames sampled from a initial 10,000-step energy minimization with MDFF forces active, followed by 400 frames sampled from the initial 40 ps of the MDFF dynamics trajectory. The structure is divided into 180 components: the 180 protein segments, each with an identical residue sequence, that make up the symmetrical virus capsid. In Figure 6a, the pronounced pattern of horizontal stripes indicates a periodic segment-to-segment variation of cross correlation values, following differences in the capsid's sym-

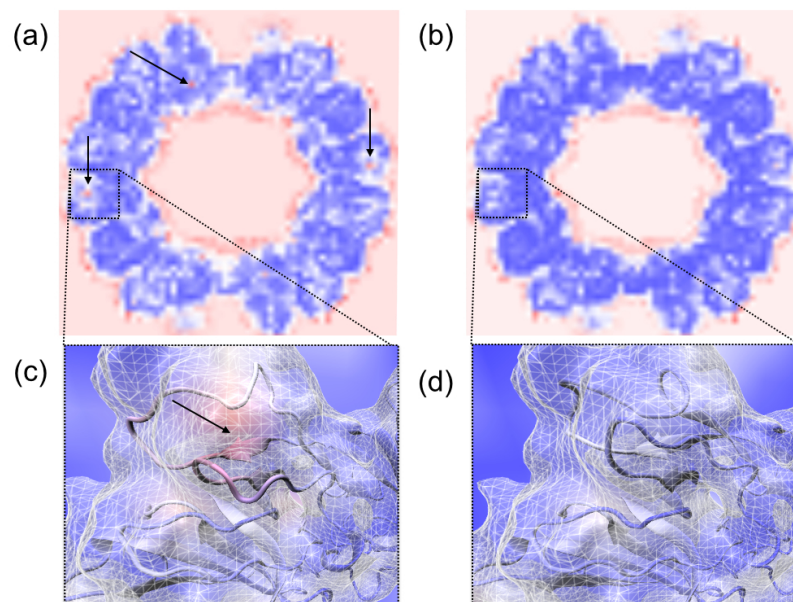


Fig. 4 Planar VolumeSlice representation of spatial cross correlation maps for the (a) first and (b) last frames of an MDFF simulation of the rabbit hemorrhagic disease virus capsid. These maps visualize local cross correlation of $8 \times 8 \times 8$ voxel regions, with red indicating poorest correlation and dark blue indicating best correlation. Regions of poorest correlation lie on the exterior and interior faces of the capsid, as would be expected. Poor-fitting regions within the subunits of the structure are seen as red spots and are marked with arrows. Panels (c) and (d) show closer views of one of these poor-fitting regions, before and after fitting, respectively. The structure is shown inside the experimental density map and is colored according to the corresponding spatial cross correlation volume around that region, to further visualize which parts of the structure have a poor fit. Panel (c) clearly shows a poorly fit region, with parts of the structure entirely outside of the density; the poor fit has been fixed by MDFF as is evident in (d).

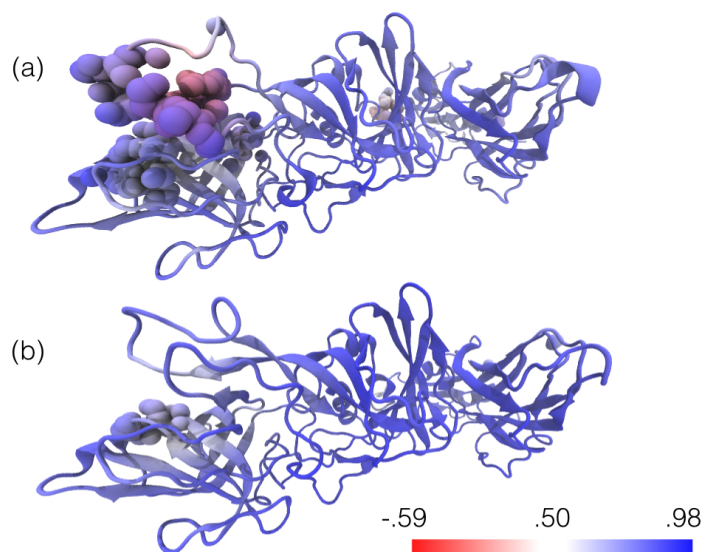


Fig. 5 One subunit of the rabbit hemorrhagic disease virus capsid before (a) and after (b) MDFF. The structure is colored according to the spatial cross correlation volume around that region. A VDW representation shows part of the structure selected as those atoms which lie inside voxels of the spatial cross correlation volume with values < 0.1 .

metrical segment arrangement, as seen in Figure 6b. Figure 6c shows relative changes observed for the individual capsid segments during the trajectory: the trajectory values are normalized by subtracting the initial cross correlation value for each segment. About half of the cross-correlation change seen in Figure 6c occurs during the initial minimization to frame 100, with much of the remaining improvement occurring quickly during the first picoseconds of the MDFF simulation, from frames 100 to 125, and minor fluctuations ceasing for the majority of segments by frame 350. Several segments can be seen to have relatively stable correlation values during both MDFF minimization and MDFF dynamics; a researcher studying the system might investigate these segments further.

Detailed analysis of very large systems with long trajectories can require a large number of individual calculations, and impose a heavy I/O burden on a computer system. We implemented a parallel *TIMELINE* cross correlation analysis using the new GPU cross correlation algorithm, for use on clusters or supercomputers. The new *TIMELINE* analysis calls VMD parallel scripting primitives⁴⁷ to perform the needed calculation, synchronization, and data reduction operations. A 10,000 frame trajectory of the 1.38 million atom RHDV capsid structure was examined. For cross correlation analysis, the capsid was divided into 720 structural components; each of the capsid's 180 525-residue segments were divided into four nearly-equal contiguous segments. The calculation was run on the NSF Blue Waters petascale supercomputer, using the XK7 hybrid GPU-accelerated compute nodes. Each XK7 node contains an AMD Opteron 6276 CPU with 32 GB of RAM and an NVIDIA Tesla K20X GPU accelerator

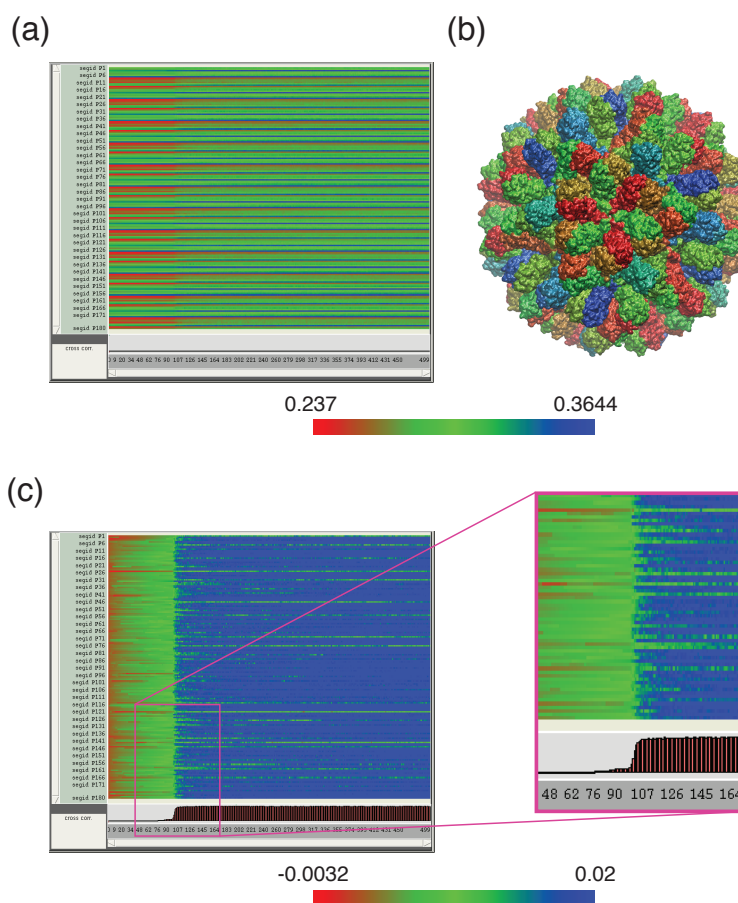


Fig. 6 Cross correlation coefficients in an rabbit hemorrhagic disease virus (RHDV) MDFF trajectory. Panel (a) contains a TIMELINE plot of the 500-frame RHDV trajectory discussed in the text, displaying the per-segment cross correlation coefficients. Panel (b) shows the RHDV structure colored by the values in the first frame (left-most column) of panel (a). Panel (c) contains a TIMELINE plot of the same trajectory as in panel (a), displaying the change from initial value of per-segment cross correlation, namely, the per-segment values in panel (a) after subtracting the value for each segment in the first frame. The bar graph at the bottom of the plot shows (in black and coral bar height) the proportion of segments at each timestep with a change in cross correlation greater than +0.01.

with 6 GB of RAM. Structure and trajectory files were striped over 160 Lustre file system object storage targets to ensure best I/O performance⁴⁷. The analysis completed in an average 19.5 minutes using 2,048 nodes, showing a $1,035\times$ speedup over a projected 336 hour (14.0 day) runtime using a single node. Using the Table 2 performance results, this would roughly correspond to a $35,800\times$ speedup over an estimated 485 day (1.3 year) runtime for Chimera running on a single node and a $142,100\times$ speedup over an estimated 1,924 day (5.3 year) runtime for the original sequential VMD CPU algorithm running on a single node.

7 Conclusions

We have presented data-parallel cross correlation algorithms that combine density map synthesis, computation of spatial cross correlation maps, and overall cross correlation between the synthetic and reference density maps, using a high performance single-pass approach on multi-core CPUs using SSE vector instructions, and on massively parallel GPUs using CUDA. The new algorithms achieve significant speedups compared to existing methods and they maintain adequate numerical precision. The single-pass algorithms achieve high performance by fusing independent algorithm steps into a single computational kernel, thereby maximizing the ratio of arithmetic operations vs. memory references, which is of greatest overall benefit to massively parallel GPU hardware. We have demonstrated the use of the new algorithms in several visualization and analysis contexts over a range of problem sizes, and we have incorporated the high performance cross correlation method into a TIMELINE multi-node parallel analysis script enabling large systems and long trajectories to be analyzed rapidly on clusters or supercomputers.

We expect that with further algorithm development, the performance of the new cross correlation algorithm can still be improved further. The simulated density map algorithm presently operates on a uniform grid that encompasses the bounding volume of an atom selection. The use of the uniform grid decomposition for the density map calculation can result in some unnecessary computation for “empty” regions of the volume that are not associated with selected atoms, since they must still participate in parallel reduction operations that yield the final correlation results. The use of the uniform grid approach provides high performance in many common cases, but it is clear that as cryo-EM imaging techniques improve and both the resolution and size of resolved biomolecular complexes increase, there will be a need for a density map algorithm that performs well for atom selections that yield sparse regions of computation. There are also several opportunities for further performance improvements in the case of trajectory analyses, such as those performed with TIMELINE, by eliminating host-GPU memory copies associated with the reference map for all but the first timestep, and by eliminating computations that are common to multiple atom selections in analyses that compute cross correlations for a large number of atom selections for each trajectory frame⁴⁷.

New data-parallel *shuffle* instructions available on recent NVIDIA GPUs could be used to accelerate the intra-block and inter-block parallel reductions that take place within the GPU algorithm. The most recent Intel and AMD processors have added new 8-element AVX and AVX2 vector instructions, offering up to a $2\times$

performance gain over SSE instructions. The greatly increased performance levels achieved by the new algorithms create many new opportunities for interactive analyses and visualizations and for interactive molecular dynamics simulations⁴⁸ using the MDFF method.

Acknowledgements

This research is part of the Blue Waters sustained-petascale computing project supported by NSF award OCI 07-25070, and “The Computational Microscope” NSF PRAC award. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications (NCSA). The authors thank Yanxin Liu for providing simulation trajectories. The authors wish to acknowledge support of the CUDA Center of Excellence at the University of Illinois and NIH funding through grants 9P41GM104601 and 5R01GM098243-02.

References

- 1 G. Zhao, J. R. Perilla, E. L. Yufenyuy, X. Meng, B. Chen, J. Ning, J. Ahn, A. M. Gronenborn, K. Schulten, C. Aiken and P. Zhang, *Nature*, 2013, **497**, 643–646.
- 2 J. Zhang, M. L. Baker, G. F. Schröder, N. R. Douglas, S. Reissmann, J. Jakana, M. Dougherty, C. J. Fu, M. Levitt, S. J. Ludtke, J. Frydman and W. Chiu, *Nature*, 2010, **463**, 379–383.
- 3 S. J. Ludtke, M. L. Baker, D.-H. Chen, J.-L. Song, D. T. Chuang and W. Chiu, *Structure*, 2008, **16**, 441–448.
- 4 X. Zhang, E. Settembre, C. Xu, P. R. Dormitzer, R. Bellamy, S. C. Harrison and N. Grigorieff, *Proc. Natl. Acad. Sci. USA*, 2008, **105**, 1867–1872.
- 5 M. L. Baker, J. Zhang, S. J. Ludtke and W. Chiu, *Nat. Protoc.*, 2010, **5**, 1697–1708.
- 6 C. F. Hryc, D.-H. Chen and W. Chiu, *Curr. Opin. Virol.*, 2011, **1**, 110–117.
- 7 R. Zhang, C. F. Hryc, Y. Cong, X. Liu, J. Jakana, R. Gorchakov, M. L. Baker, S. C. Weaver and W. Chiu, *EMBO J.*, 2011, **30**, 3854–3863.
- 8 S. Maki-Yonekura, K. Yonekura and K. Namba, *Nat. Struct. Mol. Biol.*, 2010, **17**, 417–422.
- 9 L. Cheng, J. Zhu, W. H. Hui, Z. Zhang, B. Honig, Q. Fang and Z.-H. Zhou, *J. Mol. Biol.*, 2010, **397**, 852–863.
- 10 F. Fabiola and M. S. Chapman, *Structure*, 2005, **13**, 389–400.
- 11 A. M. Roseman, *Acta Cryst. D*, 2000, **56**, 1332–1340.
- 12 R. Das and D. Baker, *Annu. Rev. Biochem.*, 2008, **77**, 363–382.
- 13 G. F. Schröder, A. T. Brunger and M. Levitt, *Structure*, 2007, **15**, 1630–1641.
- 14 C. C. Jolley, S. A. Wells, P. Fromme and M. F. Thorpe, *Biophys. J.*, 2008, **94**, 1613–1621.
- 15 F. Tama, O. Miyashita and C. L. Brooks III, *J. Struct. Biol.*, 2004, **147**, 315–326.
- 16 M. Orzechowski and F. Tama, *Biophys. J.*, 2008, **95**, 5692–5705.
- 17 M. S. Chapman, A. Trzynka and B. K. Chapman, *J. Struct. Biol.*, 2013, **182**, 10–21.
- 18 M. Topf, K. Lasker, B. Webb, H. Wolfson, W. Chiu and A. Sali, *Structure*, 2008, **16**, 295–307.
- 19 L. G. Trabuco, E. Villa, K. Mitra, J. Frank and K. Schulten, *Structure*, 2008, **16**, 673–683.
- 20 L. G. Trabuco, E. Villa, E. Schreiner, C. B. Harrison and K. Schulten, *Methods*, 2009, **49**, 174–180.
- 21 J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale and K. Schulten, *J. Comp. Chem.*, 2005, **26**, 1781–1802.
- 22 D. Wells, V. Abramkina and A. Aksimentiev, *J. Chem. Phys.*, 2007, **127**, 125101–125101–10.
- 23 E. Schreiner, L. G. Trabuco, P. L. Freddolino and K. Schulten, *BMC Bioinform*, 2011, **12**, 190.
- 24 K.-Y. Chan, J. Gumbart, R. McGreevy, J. M. Watermeyer, B. T. Sewell and K. Schulten, *Structure*, 2011, **19**, 1211–1218.
- 25 E. Villa, J. Sengupta, L. G. Trabuco, J. LeBarron, W. T. Baxter, T. R. Shaikh, R. A. Grassucci,

- P. Nissen, M. Ehrenberg, K. Schulten and J. Frank, *Proc. Natl. Acad. Sci. USA*, 2009, **106**, 1063–1068.
- 26 J. Gumbart, L. G. Trabuco, E. Schreiner, E. Villa and K. Schulten, *Structure*, 2009, **17**, 1453–1464.
- 27 T. Becker, S. Bhushan, A. Jarasch, J.-P. Armache, S. Funes, F. Jossinet, J. Gumbart, T. Mielke, O. Berninghausen, K. Schulten, E. Westhof, R. Gilmore, E. C. Mandon and R. Beckmann, *Science*, 2009, **326**, 1369–1373.
- 28 B. Seidelt, C. A. Innis, D. N. Wilson, M. Gartmann, J.-P. Armache, E. Villa, L. G. Trabuco, T. Becker, T. Mielke, K. Schulten, T. A. Steitz and R. Beckmann, *Science*, 2009, **326**, 1412–1415.
- 29 L. G. Trabuco, E. Schreiner, J. Eargle, P. Cornish, T. Ha, Z. Luthey-Schulten and K. Schulten, *J. Mol. Biol.*, 2010, **402**, 741–760.
- 30 J. Gumbart, E. Schreiner, L. G. Trabuco, K.-Y. Chan and K. Schulten, *Molecular Machines in Biology*, Cambridge University Press, 2011, ch. 8, pp. 142–157.
- 31 J. Frauenfeld, J. Gumbart, E. O. van der Sluis, S. Funes, M. Gartmann, B. Beatrix, T. Mielke, O. Berninghausen, T. Becker, K. Schulten and R. Beckmann, *Nat. Struct. Mol. Biol.*, 2011, **18**, 614–621.
- 32 X. Agirrezabala, E. Schreiner, L. G. Trabuco, J. Lei, R. F. Ortiz-Meoz, K. Schulten, R. Green and J. Frank, *EMBO J.*, 2011, **30**, 1497–1507.
- 33 W. Li, L. G. Trabuco, K. Schulten and J. Frank, *Proteins: Struct., Func., Bioinf.*, 2011, **79**, 1478–1486.
- 34 J. Hsin, J. Gumbart, L. G. Trabuco, E. Villa, P. Qian, C. N. Hunter and K. Schulten, *Biophys. J.*, 2009, **97**, 321–329.
- 35 M. K. Sener, J. Hsin, L. G. Trabuco, E. Villa, P. Qian, C. N. Hunter and K. Schulten, *Chem. Phys.*, 2009, **357**, 188–197.
- 36 D. Vasishtan and M. Topf, *J. Struct. Biol.*, 2011, **174**, 333–343.
- 37 R. Diamond, *Acta Cryst. A*, 1971, **27**, 436–452.
- 38 T. A. Jones and L. Liljas, *Acta Cryst. A*, 1984, **40**, 50–57.
- 39 P. L. Stewart, S. D. Fuller and R. M. Burnett, *EMBO J.*, 1993, **12**, 2589–2599.
- 40 W. Wriggers, *Biophysical Reviews*, 2010, **2**, 21–27.
- 41 W. Wriggers, *Acta Cryst. D*, 2012, **68**, 344–351.
- 42 E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng and T. E. Ferrin, *J. Comp. Chem.*, 2004, **25**, 1605–1612.
- 43 S. J. Ludtke, P. R. Baldwin and W. Chiu, *J. Struct. Biol.*, 1999, **128**, 82–97.
- 44 M. S. Chapman, *Acta Cryst. A*, 1995, **51**, 69–80.
- 45 P. Chacón and W. Wriggers, *J. Mol. Biol.*, 2002, **317**, 375–384.
- 46 W. Humphrey, A. Dalke and K. Schulten, *J. Mol. Graphics*, 1996, **14**, 33–38.
- 47 J. E. Stone, B. Isralewitz and K. Schulten, Proceedings of the XSEDE Extreme Scaling Workshop, 2013.
- 48 J. E. Stone, J. Gullingsrud, P. Grayson and K. Schulten, 2001 ACM Symposium on Interactive 3D Graphics, New York, 2001, pp. 191–194.
- 49 J. Frank, *Three-dimensional electron microscopy of macromolecular assemblies*, Oxford University Press, New York, 2006.
- 50 W. Kahan, *Commun. ACM*, 1965, **8**, 40–.
- 51 Y. He and C. H. Q. Ding, ICS '00: Proceedings of the 14th international conference on Supercomputing, New York, NY, USA, 2000, pp. 225–234.
- 52 D. H. Bailey, *Comput. in Sci. and Eng.*, 2005, **07**, 54–61.
- 53 J. E. Stone, J. Saam, D. J. Hardy, K. L. Vandivort, W. W. Hwu and K. Schulten, Proceedings of the 2nd Workshop on General-Purpose Processing on Graphics Processing Units, ACM International Conference Proceeding Series, New York, NY, USA, 2009, pp. 9–18.
- 54 M. Krone, J. E. Stone, T. Ertl and K. Schulten, EuroVis - Short Papers 2012, 2012, pp. 67–71.
- 55 E. Roberts, J. E. Stone and Z. Luthey-Schulten, *J. Comp. Chem.*, 2013, **34**, 245–255.
- 56 J. E. Stone, K. L. Vandivort and K. Schulten, Proceedings of the 8th International Workshop on Ultrascale Visualization, New York, NY, USA, 2013, pp. 6:1–6:8.
- 57 J. Nickolls, I. Buck, M. Garland and K. Skadron, *ACM Queue*, 2008, **6**, 40–53.
- 58 A. Sali and T. L. Blundell, *J. Mol. Biol.*, 1993, **234**, 779.
- 59 X. Wang, F. Xu, J. Liu, B. Gao, Y. Liu, Y. Zhai, J. Ma, K. Zhang, T. S. Baker, K. Schulten,

-
- D. Zheng, H. Pang and F. Sun, *PLoS Pathog.*, 2013, **9**:e1003132.
60 K.-Y. Chan, L. G. Trabuco, E. Schreiner and K. Schulten, *Biopolymers*, 2012, **97**, 678–686.