Structural Dynamics

Liquid-like and rigid-body motions in molecular-dynamics simulations of a crystalline protein

Cite as: Struct. Dyn. **6**, 064704 (2019); https://doi.org/10.1063/1.5132692
Submitted: 21 October 2019 . Accepted: 19 November 2019 . Published Online: 18 December 2019

David C. Wych, James S. Fraser, David L. Mobley, and Michael E. Wall

Note: This article is part of the Special Issue: Transactions from the 69th Annual Meeting of the American Crystallographic Association: Data Best Practices: Current State and Future Needs.

This paper was selected as Featured

Structural Dynamics ARTICLE scitation.org/journal/sdy

Liquid-like and rigid-body motions in molecular-dynamics simulations of a crystalline protein

Cite as: Struct. Dyn. 6, 064704 (2019); doi: 10.1063/1.5132692
Submitted: 21 October 2019 · Accepted: 19 November 2019 ·
Published Online: 18 December 2019

David C. Wych,[1,2] James S. Fraser,[3] David L. Mobley,[1,4] and Michael E. Wall[2,a)]

AFFILIATIONS

[1]Department of Pharmaceutical Sciences, University of California, Irvine, Irvine, California 92697, USA
[2]Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA
[3]Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, San Francisco, California 94143, USA
[4]Department of Chemistry, University of California, Irvine, Irvine, California 92697, USA

Note: This article is part of the Special Issue: Transactions from the 69th Annual Meeting of the American Crystallographic Association: Data Best Practices: Current State and Future Needs.
[a)]**Author to whom correspondence should be addressed:** mewall@lanl.gov

ABSTRACT

To gain insight into crystalline protein dynamics, we performed molecular-dynamics (MD) simulations of a periodic $2 \times 2 \times 2$ supercell of staphylococcal nuclease. We used the resulting MD trajectories to simulate X-ray diffraction and to study collective motions. The agreement of simulated X-ray diffraction with the data is comparable to previous MD simulation studies. We studied collective motions by analyzing statistically the covariance of alpha-carbon position displacements. The covariance decreases exponentially with the distance between atoms, which is consistent with a liquidlike motions (LLM) model, in which the protein behaves like a soft material. To gain finer insight into the collective motions, we examined the covariance behavior within a protein molecule (intraprotein) and between different protein molecules (interprotein). The interprotein atom pairs, which dominate the overall statistics, exhibit LLM behavior; however, the intraprotein pairs exhibit behavior that is consistent with a superposition of LLM and rigid-body motions (RBM). Our results indicate that LLM behavior of global dynamics is present in MD simulations of a protein crystal. They also show that RBM behavior is detectable in the simulations but that it is subsumed by the LLM behavior. Finally, the results provide clues about how correlated motions of atom pairs both within and across proteins might manifest in diffraction data. Overall, our findings increase our understanding of the connection between molecular motions and diffraction data and therefore advance efforts to extract information about functionally important motions from crystallography experiments.

NOMENCLATURE

LLM	liquid-like motions
MD	molecular dynamics
pdTp	thymidine-3′-5′-bisphosphate
RBM	rigid-body motions

INTRODUCTION

Macromolecular crystals consist of many copies of a large molecule (or molecules) packed into a lattice of repeating units. Crystals are often illustrated using identical repeating units; however, in real crystals, each copy of a molecule can adopt a somewhat different structure as long as the overall order is maintained. Although structural variations occur in small-molecule crystals,[1] the problem of conformational heterogeneity is especially important for macromolecular crystals, which have many more degrees of freedom and a high solvent content.[2,3] Understanding the structural variations in crystals can potentially be used for increasing the accuracy of crystal structure models[4] and for developing crystallography as a tool for characterizing conformational heterogeneity and dynamics in structural biology.[5]

In X-ray crystallography (and similarly for neutron and electron crystallography), the details of the molecules' conformations and their packing in the crystal lattice leave signatures in the diffraction pattern. Most of our current understanding of conformational variation in crystals comes from the analysis of Bragg reflections, which are the sharp peaks in the diffraction pattern. The Bragg peaks are traditionally modeled using an average picture of the repeating unit, or unit cell, which contains the mean electron density over all unit cells illuminated by the beam. The most common model multiplies each atomic form factor by an atomic displacement parameter (also known as a Debye-Waller factor, B-factor, or thermal factor) corresponding to a 3D Gaussian distribution of each atom's displacements.

Modern Bragg analysis methods are producing richer descriptions of conformational variations. For example, anisotropic displacements of groups of atoms can be modeled with a small number of parameters using the Translation Libration Screw (TLS) model,[6] which can be used as a supplement or substitute to refining individual atomic displacement parameters.[7] More elaborate models of conformational heterogeneity can also be employed,[8] including incorporating local alternative conformations in multiconformer models[9,10] or generating multiple models using hybrid molecular dynamics ensemble refinement that collectively satisfy the data.[10] These models can imply distinct collective motions that would leave signatures in the spatial correlations of electron density variations. Therefore, even if more elaborate models of motion improve agreement with the Bragg data, they would need to be validated to determine whether they describe what is actually happening in the crystal.[11] Indeed, distinct TLS refinements with different implied collective motions can yield equivalent agreement with the Bragg data.[12]

The degeneracy of models with respect to the mean electron density fortunately does not extend to spatial correlations in electron density variations. Models with different correlations lead to distinct patterns of diffuse scattering that appear as intensity beneath and between the Bragg peaks in diffraction images. Recent years have seen a renaissance in methods for processing and modeling diffuse scattering.[2,3,13] However, studies of diffuse scattering have differed in their conclusions, both about the degree to which the signal can be explained by various models of correlated motion and about the importance of different types of motion in determining the conformational variations in protein crystals.[2,14–17] Some studies find evidence for liquidlike motions (LLM), which model the crystal contents as a soft material with correlated displacements on a characteristic length scale.[18] Others find evidence for rigid-body motions (RBM),[15] which could, in principle, be modeled by descriptions such as TLS models.[12] Ensemble models, in which a limited set of representative structures are selected from a full conformational distribution,[19] also have been studied and found to be lacking.[14,15] A potential problem with ensemble models is that they exhibit exaggerated correlations due to the absence of finer-scale structure variations.[20] More sophisticated models might be able to better capture the correlated motions within the crystal.[21] Considering the true crystalline context by including neighboring unit cells in the calculations can increase agreement of simple models like LLM.[14]

Whereas each of the above-mentioned models of diffuse scattering explicitly represents only a subset of the possible structural variations, molecular-dynamics (MD) simulations of crystalline proteins provide an all-atom picture of a wide variety of available motions.[22–25] Early MD simulations of diffuse scattering were hindered by the short duration (<1 ns), which resulted in limited sampling of the conformational ensemble and calculations that were not reproducible across different runs.[26] Agreement between experimental and predicted diffuse scattering of crystalline staphylococcal nuclease increased as the simulation duration increased to 10 ns[27] and then even more as it was increased to 1 μs.[25] A recent simulation of staphylococcal nuclease extended the simulation volume from a single periodic unit cell to a 2 × 2 × 2 supercell, leading to a further increase in agreement with the experimental data.[28] This result parallels the emerging theme from the LLM work,[14] which highlighted the importance of considering more than just a single unit cell to generate strong agreement with experimental data.[29] MD provides a full description of all atoms in the system but has a high computational cost relative to the simpler models, such as LLM and RBM. Because simpler models can potentially be incorporated into joint X-ray Bragg and diffuse scattering model refinement, it is interesting to investigate the degree to which the motions described by simpler models, such as the LLM and RBM models, appear in the increasingly accurate MD simulations that are now achievable. It is worth noting that while the LLM and RBM models include parameters explicitly fit to match diffuse scattering patterns, MD models utilize no such fit, making this comparison particularly interesting.

Here, we assess the degree to which LLM and RBM behaviors are present in MD simulations of crystalline staphylococcal nuclease. We performed new MD simulations of crystalline staphylococcal nuclease and confirmed that their agreement with diffuse scattering data is similar to previous studies. We then computed covariance matrices of Cα atom displacements from the resulting MD trajectories and analyzed the dependence of the covariance on the average separation between atoms, which are covarying. The covariance behavior is well fit by an exponential decrease with distance, supporting a LLM model. However, when the analysis is restricted to atom pairs that lie within the same protein, the behavior is consistent with a combination of LLM and RBM. Comparison of results obtained using AMBER vs CHARMM force fields reveals some differences in the simulations and covariance behavior. Our results indicate that LLM behavior of global dynamics is present in MD simulations of a protein crystal. They also show that RBM behavior is detectable in the simulations but that it is subsumed by the LLM behavior when adding atom pairs that cross protein boundaries. Finally, they provide clues about how correlated motions of atom pairs both within and across proteins might manifest in the diffuse scattering data. Overall, our findings increase our understanding of the connection between molecular motions and diffuse scattering data and therefore advance efforts to extract information about functionally important motions from crystallography experiments.

RESULTS

Molecular dynamics simulations

In this study, we sought to investigate the connection between MD simulations and other, less detailed models of crystalline dynamics, like the LLM model. For the MD simulations, we used a model of crystalline staphylococcal nuclease, consisting of a periodic box consisting of 2 × 2 × 2 unit cells (Fig. 1; Methods). Solution state simulations commonly are performed using a NPT ensemble; however, as we wished to compare our simulations to diffraction data, we used a NVT

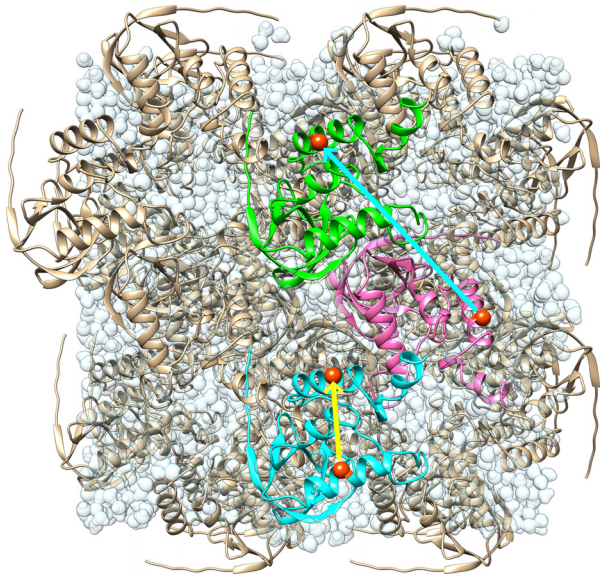

FIG. 1. Illustration of the model used for molecular dynamics simulations. There are 32 copies of the protein rendered using ribbons in a periodic box of 2 × 2 × 2 unit cells. The yellow arrow indicates an atom pair that lies entirely within the blue protein, pointing from residue 61 to residue 131 (corresponding to the "within" or "intraprotein" analysis). The cyan arrow indicates an atom pair that spans across proteins, pointing from residue 131 in the magenta to residue 128 in the green protein (corresponding to the "across" or "interprotein" analysis). Water molecules are rendered in light, transparent blue, giving the appearance of connected droplets. The image was created using UCSF "Chimera."[30]

ensemble, maintaining the consistency of the system with the experimentally determined Bragg lattice during the course of the simulation. To determine the sensitivity of the results to the force field, we used both AMBER 14SB and CHARMM 27 force fields in GROMACS.

Consistent with previous studies,[25,28] which used similar methods to the present study, after initial solvation and minimization, during the first equilibration step, the pressure of the systems was large and negative: −1439 ± 39 bar for the AMBER simulation, and −1795 ± 252 bar for the CHARMM simulation (as reported by "gmx energy)." To bring the system to atmospheric pressure, additional water molecules were added iteratively, with intervening rounds of NVT equilibration, until the mean system pressure was in the range of −100 to 100 bar (Methods). The number of water molecules added was similar for both systems (17 557 waters for AMBER and 17 138 waters for CHARMM).

After the iterative solvation steps, unrestrained MD simulations were carried out for 600 ns. The potential energy drifted during the first ∼100 ns the simulation, after restraints were released, and remained relatively stable thereafter. To ensure that the drift did not influence the results of our analysis of the trajectory, the initial 200 ns section was ignored, and only the last 400 ns section was analyzed. At the 200 ns time point, the root mean squared deviation (RMSD) of the Cα atom coordinates between the crystal structure and the MD model after translational superposition was 2.7 Å for the AMBER simulation and 2.6 Å for the CHARMM simulation. (We note that RMSDs from solution state simulations are usually computed after performing both translational and rotational alignment of individual proteins, whereas these RMSDs were computed after performing translational alignment of the entire supercell). The RMSD slowly increased through the 600 ns time point, at which it was 3.1 Å for the AMBER simulation and 2.9 Å for the CHARMM simulation (supplementary Fig. S1). The B-factors predicted by the atomic fluctuations (not shown) were consistent with previous studies of the same system,[28] and with the experimentally determined B factors.

Simulated diffuse intensity

To determine the agreement of the simulations with diffuse scattering data, we computed diffuse intensities from MD trajectories and computed the Pearson correlation coefficient between the simulated and experimental intensities (Methods). Simulated diffuse diffraction images computed from the MD simulations and experimental data are compared in Fig. 2. The Pearson correlation between the simulated and experimental total 3D diffuse intensities was 0.9 or higher for all simulations, similar to that found in a previous study utilizing the same approach.[28] The correlation of the anisotropic component of the intensity, computed by subtracting an interpolated radial average,[28] was 0.58 (AMBER) and 0.63 (CHARMM) for the anisotropic intensity, which is lower than the value of 0.68 previously reported in Ref. 28. We reanalyzed the data in Ref. 28 and obtained the same correlation of 0.68, confirming that the difference is attributed to the simulations rather than the analysis workflow.

To gain finer-grained insight into the agreement between the simulated and experimental diffuse intensities, we analyzed the trajectory in 100 ns sections (supplementary Fig. S2). The correlation computed using each section is between 0.53 and 0.58 which is lower than the 0.58 and 0.63 values obtained using the diffuse intensity accumulated for 400 ns. The cumulative agreement is also sensitive to whether the diffuse intensity is accumulated coherently or incoherently across the individual 100 ns sections (see Methods for definition of coherent vs incoherent accumulation): when accumulated coherently, the correlation is 0.54 (AMBER) 0.58 (CHARMM) which is lower than the values 0.58 (AMBER) and 0.63 (CHARMM) when accumulated incoherently.

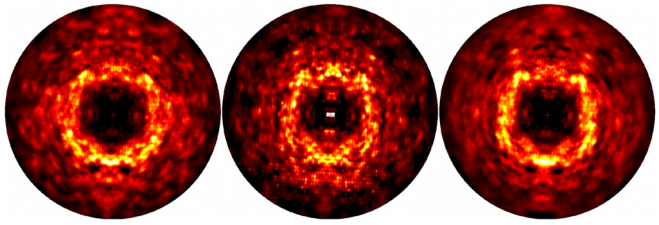

FIG. 2. Simulated diffraction images derived from MD simulations and 3D experimental diffuse data. 2 × 2 × 2 sampling. The images are truncated at 1.8 Å. To make the anisotropic features more visible, the 3D diffuse intensities have had the minimum intensity value subtracted in constant resolution shells prior to generating these images. Left: image derived from AMBER force-field simulation. Center: image derived from experimental data. Right: image derived from CHARMM force-field simulation. The diffuse intensity for MD simulations was accumulated incoherently across 100 ns sections of the trajectory in the time range 200–600 ns after relaxing restraints. Both the AMBER and CHARMM MD simulations show similar diffuse features to those in the experimentally derived images, for example, the strong intensity in the ring and cloudy features at higher resolution. The images were displayed using the heat map mode of ADXV.[31]

Liquidlike motion behavior of all Cα atom pairs

To assess whether the MD simulations produced behavior that is consistent with the LLM model, we computed atom displacement covariance matrices (Methods). The covariance matrices are needed because the key assumption of the LLM model is that the covariance matrix elements connecting any two atoms i and j in the crystal are simply proportional to $e^{-r_{ij}/\gamma}$, where r_{ij} is the distance between the atoms, and γ is the characteristic length scale of the correlations.[18] To make the calculations and analysis manageable computationally and to obtain a coarse-grained picture of the covariance, we restricted the analysis to Cα atoms.

The full covariance matrix of Cα displacements for our system is a 14 304 × 14 304 square matrix with each row or column corresponding to one of three cartesian coordinates of each of 149 Cα atoms in each of 32 proteins in the model. To perform our analyses, we replaced the 3 × 3 submatrix for each atom pair with its trace, leaving a 4768 × 4768 square symmetric matrix. In this form, the diagonal elements correspond to the mean squared deviation (MSD) for each atom, and the off-diagonal elements correspond to the trace of the covariance of the displacements of each atom pair. The full covariance matrix contains regions of both positive and negative covariance, with the strongest positive values being in blocks about the diagonal (supplementary Fig. S3), corresponding to atom pairs that fall within a protein (as connected by the yellow line in Fig. 1).

To determine the dependence of the matrix elements coupling atoms i and j on the distance between the atoms r_{ij}, we computed the distance between each of the atom pairs and divided the distance range into 50 even bins. We then calculated the mean and standard error of the covariance within each of the bins (Fig. 3; Methods). For each simulation, at the lowest distance there is a single bin with high covariance compared to the other bins: in the AMBER case, a point at 1.67 Å with covariance 15.5 Å^2 ± 1.84 Å^2 is not shown as it is out of range in y; in the CHARMM case, a point at 1.47 Å with covariance 21.8 Å^2 ± 3.06 Å^2 is not shown as it is out of range in y. Beyond 5 Å (Fig. 3), the covariance is much lower (20-fold lower than in the nearest bin below) and shows a more gradual decrease with distance, falling from a value of 0.30 Å^2 at 5 Å, crossing below zero beyond about 40 Å to a minimum value of either −0.02 (AMBER) or −0.03 (CHARMM) Å^2 beyond about 50 Å, and rising to a value closer to zero beyond about 80 Å (AMBER) or 60 Å (CHARMM).

To assess whether these plots display exponential decay behavior, the values of the covariance $C(r)$ were fit to the function $C(r) = ae^{-r/\gamma} + b$, where r is the distance between atoms, in the range r between 5 Å and 55.5 Å. (We found that the exponential fit was poor without adding the constant b, and so we added it.) For the AMBER simulation, the fit yielded $a = 0.79 \pm 0.01$ Å^2, $\gamma = 11.0 \pm 0.1$ Å, and $b = -0.022 \pm 0.001$ Å^2. For the CHARMM simulation, the fit yielded $a = 0.94 \pm 0.02$ Å^2, $\gamma = 11.1 \pm 0.2$ Å, and $b = -0.029 \pm 0.001$ Å^2. The constant offset at long distances is consistent with an earlier simulation,[27] which postulated that it might be an artifact of translational alignment of the MD trajectory snapshots—a necessary step before computing the covariance matrix (Methods). Figure 3 shows that the fit overlaps the computed covariances in the region below 60 Å, and therefore displays exponential decay behavior, which is consistent with the assumption of the LLM.

Combination of liquidlike and rigid-body motion behaviors within proteins

As noted in Liquidlike motion behavior, the strongest positive values of the covariance matrix were in blocks along the diagonal (supplementary Fig. S3). These blocks correspond to Cα atom pairs that lie within the same protein (yellow line in Fig. 1), which we refer to as intraprotein or "within-protein" atom pairs. The rest of the covariance matrix corresponds to atom pairs that cross protein boundaries, which we refer to as interprotein or "across-protein" atom pairs (cyan line in Fig. 1). (The subsets of intraprotein and interprotein Cα atom pairs are complementary with respect to the set of all Cα atom pairs.) We wondered whether the LLM behavior observed for all Cα atom pairs also would apply individually to these subsets. We therefore computed the covariance vs distance for each. We also wished to focus our attention on the more stable region of the protein and therefore eliminated residues at the extreme N- and C-terminal Cα atoms from our calculations (Methods).

The shape of the curve for interprotein atom pairs (Fig. 4) is similar to that for all Cα atom pairs (Fig. 3). In the case of the AMBER simulation, a point at 3.3 Å with covariance −1.77 Å^2 ± 1.23 Å^2 is not shown as it is out of range in y. For AMBER, the best-fit exponential decay has $a = 0.42 \pm 0.02$ Å^2, $\gamma = 14.3 \pm 0.4$ Å, and $b = 0.025 \pm 0.001$ Å^2; for CHARMM the best-fit has $a = 0.55 \pm 0.02$ Å^2, $\gamma = 13.4 \pm 0.3$ Å,

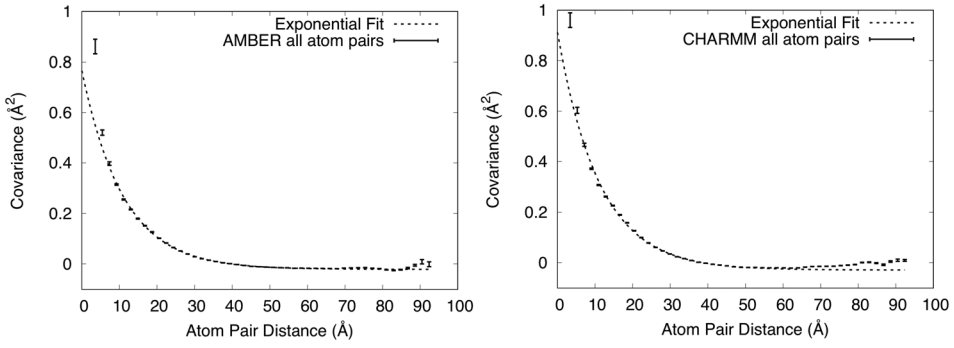

FIG. 3. Dependence of covariance on distance for all Cα atom pairs. Mean values of covariance ± standard errors are shown using vertically oriented error bars. Exponential fits to the values in the 5–55.5 Å range are shown using dashed lines. The upper y-range is truncated at 1 Å^2, showing the full range of the exponential fit but excluding one high covariance value at very short distance from each panel. Left: AMBER force field. The exponential fit has a decay length of 11.0 Å. Right: CHARMM force field. The exponential fit has a decay length of 11.1 Å.

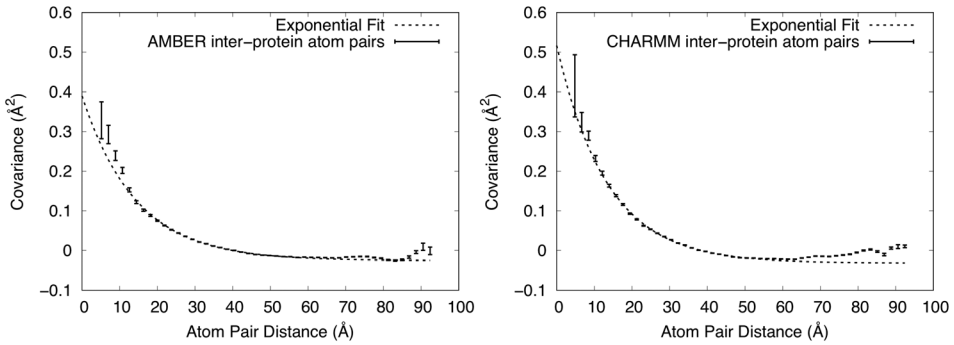

FIG. 4. Dependence of covariance on distance for just interprotein Cα atom pairs. Mean values of covariance ± standard errors are shown using vertically oriented error bars. Exponential fits to the values in the 5–55.5 Å range are shown using dashed lines. The upper y-range is truncated at 0.6 Å^2. Left: AMBER force field. The shortest distance point is not shown as it is out of range. The exponential fit has a decay length of 14.3 ± 0.4 Å. Right: CHARMM force field. The exponential fit has a decay length of 13.4 ± 0.3 Å. The exponential fits are fairly good, but not as good as in Fig. 3.

and $b = 0.032 \pm 0.001$ Å^2. The values at a short distance are smaller in the interprotein analysis than in the analysis of all atom pairs. In the case of the AMBER simulation, the exponential fit deviates from the covariance values in the region below 10 Å (Fig. 4). The values of a are smaller than those for all atom pairs, supporting the observation that the values at a short distance are smaller. The values of γ are larger than those for all atom pairs, indicating that correlations extend to a longer length scale. The values of b are similar to the values for all atom pairs.

In contrast to the interprotein atom pairs, the shape of the curve for atom pairs within proteins (Fig. 5) is very different (note that the x-axis only extends to ∼42.5 Å, while retaining the number of bins at 50). In the AMBER case, a MD point at 2.90 Å with covariance 1.64 Å^2 ± 0.38 Å^2 is not shown as it is out of range in y. In the case of the CHARMM simulation, a MD point at 3.2 Å with covariance 2.85 Å^2 ± 0.23 Å^2 is not shown as it is out of range in y. The values still decrease with increasing distance, but the curvature is lower near 5 Å. Moreover, the behavior is almost linear above 20 Å, crossing zero at about 38 Å, and decreasing to about −0.1 Å^2 at the longest distance.

In seeking an explanation for the linear behavior, we reasoned that rigid-body rotations of individual proteins should give rise to a decreasing covariance with distance. For example, during a rotation through the center of mass, nearby atoms on the surface would tend to move together, giving rise to a positive covariance, and atoms at remote locations on the surface would tend to move in opposite directions, giving rise to a negative covariance. We postulated that this might lead to a decreasing covariance with distance that is positive at short distances and becomes negative at long distances. Adding a rigid-body translation after the rotation would lead to a uniform positive covariance within the protein, shifting the curve up and the zero crossing to longer distances.

To test this idea, we generated an ensemble of snapshots displaying varying degrees of RBM. For each snapshot, three Euler angles and a translational shift were drawn from a normal distribution, and rigid coordinate transformations were applied to a single staphylococcal nuclease protein from the MD model. The widths of the distributions were chosen to be on a par with the magnitude of motion observed in our MD simulations. The covariance matrix was computed from the

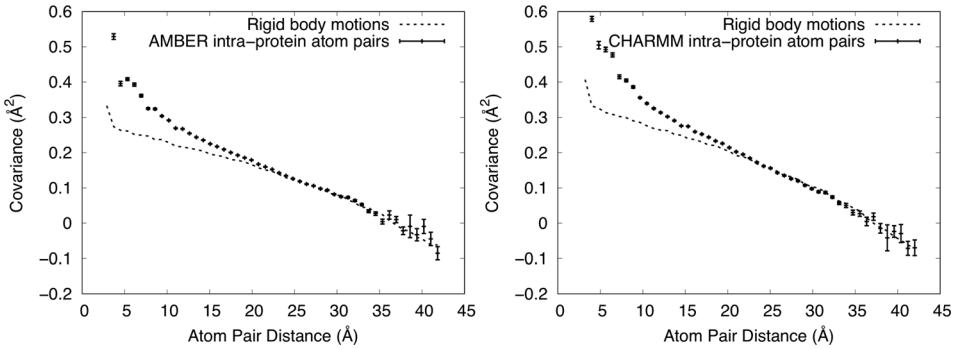

FIG. 5. Dependence of covariance on distance for only intraprotein Cα atom pairs. Mean values of covariance ± standard errors from the MD simulations are shown using vertically oriented error bars. Values computed from RBM models are shown using dashed lines (standard errors are O(10^{-5}) Å^2 and are therefore not shown). The upper y-range is truncated at 0.6 Å^2, showing the full range of the exponential fit but excluding one high covariance value at the shortest distance from the MD values in each panel. Left: AMBER force field. The RBM model has a width (SD) 0.95° for the angular distribution and 0.24 Å for the translational distribution. Right: CHARMM force field. The RBM model has a width (SD) 1.05° for the angular distribution and 0.27 Å for the translational distribution. The MD simulations are well modeled using rigid-body translations and rotations for distances greater than 20 Å.

snapshots, and the distance dependence was analyzed in the same manner as for the MD trajectories.

We adjusted the standard deviation (SD) of the angular and translational distributions to optimize the visual agreement with the MD simulation in the long-distance part of the curve. In the case of the AMBER simulation, the final SD used for the angular distribution was 0.95°, and the SD of the translational distribution was 0.24 Å. For the CHARMM simulation, the SD of the angular distribution was 1.05°, and the SD of the translational distribution was 0.27 Å. The model tracks MD simulation results in the region above about 20 Å (Fig. 4), indicating that the MD covariance in this long-distance region is consistent with RBM.

To explain the behavior in the region below 20 Å, we subtracted the RBM plots from the corresponding MD simulation plots in Fig. 5, yielding the residual shown in Fig. 6. In the AMBER case, a point at 2.9 Å with covariance 1.31 Å^2 is not shown as it is out of range in y. In the CHARMM case, a MD point at 3.2 Å with covariance 2.45 Å^2 is not shown as it is out of range in y. As the residual plots resemble an exponential decay, we again fit them to the function $C(r) = ae^{-r/\gamma} + b$ in the region $r > 5$ Å. For the AMBER simulation, the fit yielded $a = 0.37 \pm 0.02$ Å^2, $\gamma = 5.7 \pm 0.2$ Å, $b = 0$ Å^2 to within error; for CHARMM, the fit yielded $a = 0.46 \pm 0.03$ Å^2, $\gamma = 5.7 \pm 0.3$ Å, and $b = -0.002 \pm 0.001$ Å^2. The fit confirms that the residual is consistent with an exponential decay, but with a much shorter length scale than for the interprotein atom pairs.

Additional insight into the LLM behavior comes from comparing the values of γ and a obtained from the MD analysis with the values obtained by fitting a LLM model to coarsely sampled (one point per Miller index) experimental diffuse scattering data (Methods). The refined LLM model had a Pearson correlation coefficient with the anisotropic data of 0.73, with $\gamma = 6.5$ Å and $\sigma = 0.41$ Å (the agreement with the total diffuse data, as opposed to the anisotropic data, is poor, as the LLM model does not include solvents). A comparison of simulated diffraction images from the model and data is shown in supplementary Fig. S4. The value of γ is closer to the value for the intraprotein analysis (5.5 Å) than for the interprotein (11.5 Å) or all-atom (12 Å) analysis. The value of σ corresponds to a MSD of $3 \times (0.41 \text{ Å})^2 = 0.50$ Å^2, which is comparable to the values of a obtained in the intraprotein analysis (0.42 Å^2 for AMBER and 0.48 Å^2 for CHARMM) and to the values for the interprotein analysis (0.55 Å^2 for AMBER and 0.66 Å^2 for CHARMM). This comparison indicates that the coarsely sampled diffuse scattering data exhibit both a length scale of correlations and amplitude of motion that are most consistent with the intraprotein atom pairs in the MD simulation (see Discussion).

DISCUSSION

Our crystalline MD simulations of staphylococcal nuclease reveal a consistency with the LLM model: the distance dependence of the covariance of Cα displacements follows an exponential decay. This finding indicates that LLM behavior is present in a more realistic, highly detailed, all-atom description of the dynamics. It also provides a rationale for why the LLM model, which uses only a few parameters, can provide a reasonable explanation of diffuse scattering data, which depends on the atomic details of the structure variations.

The atomic details contained in the MD also allow us to examine the covariance behavior within a protein molecule (intraprotein) and between different protein molecules (interprotein). Consistent with the overall LLM model, the distance dependence of the covariance between interprotein atom pairs appears exponential, albeit with deviations below 10 Å [Fig. 4]. The best-fit values of γ are 14.3 Å for the AMBER simulation and 13.4 Å for the CHARMM simulation, which are somewhat longer than the value of 11 Å for all protein pairs. If the subset of interprotein atom pairs dominates the statistics of all atom pairs, then it should be most important in determining the LLM behavior that was observed for all atom pairs. Indeed, beyond a distance of 12 Å, the number of interprotein atom pairs sharply climbs above the number of intraprotein atom pairs (supplementary Fig. S5), supporting this notion. This finding is consistent with earlier work of Peck *et al.*[14] showing that including interactions across molecular boundaries improves agreement with the anisotropic diffuse scattering signal.

In contrast to the interprotein atom pairs, the covariance behavior for just the intraprotein atom pairs deviates from a LLM model. For these atom pairs, the behavior is described by a mixture of RBM and other contributions. The long-range behavior (above 20 Å) is almost exclusively explained by RBM, the midrange behavior (5-20 Å) is dominated by RBM, and in the short-range region (below 5 Å), the RBM model accounts for only a minority of the covariance. The

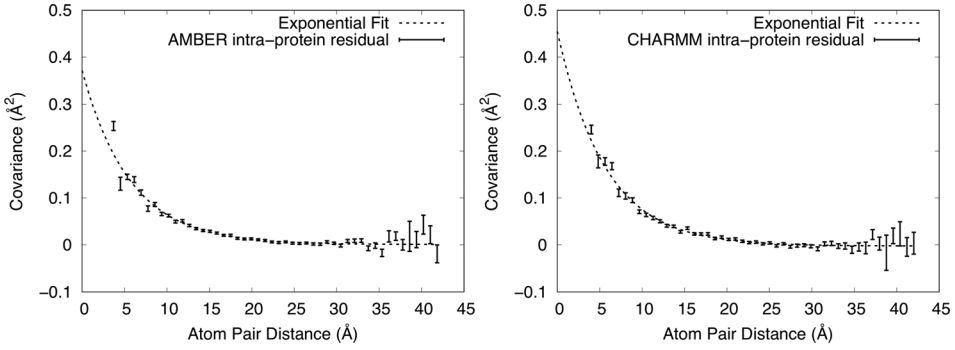

FIG. 6. Residual covariance computed from the MD simulations, after subtracting the covariance values of the RBM model (see Fig. 5). Mean values of covariance ±standard errors from the MD simulations are shown using vertically oriented error bars. Exponential fits to the values in the range above 5 Å are shown using dashed lines. The upper y-range is truncated at 0.5 Å^2, showing the full range of the exponential fit but excluding one high covariance value at the shortest distance from each panel. Left: AMBER force field. Right: CHARMM force field. Both residuals are well fit by an exponential with the same decay length of 5.7 Å.

observation of RBM here is reminiscent of ssNMR studies of ubiquitin crystals[32,33] in which MD simulations were used to explain the crystal dynamics; in that study, 3–5° rocking motions were observed via rotational alignment of proteins from the MD trajectory. We note, however, that 3–5° is much larger than the 1° SD that explains the covariance behavior in Fig. 5. In addition, the present results are consistent with our previous analysis of rigid-body rotations based on rotational alignment of proteins from the MD trajectory;[28] that analysis found 1–2° SDs of Euler angles and indicated that rigid-body rotations account for a minority of the atom displacements in staphylococcal nuclease crystalline MD simulations.

After subtracting the RBM contribution from the intraprotein covariance plot, the residual is well fit by an exponential decay, except below 5 Å. The decay length of the fit is 5.7 Å, which is substantially shorter than the length found for all atom pairs (11 Å) or just the interprotein atom pairs (13.4 Å and 14.3 Å). As the decay length differs, it is possible that the interprotein and intraprotein LLM behaviors have different origins in the MD simulations. As the intraprotein LLM behavior only is apparent after the RBM contribution has been subtracted, it is unlikely to involve a substantial RBM component; however, it is possible that the interprotein LLM behavior includes a component that is due to the coupling of RBM across protein boundaries.[34] In addition, our findings do not rule out the possibility that the LLM behavior includes coupled rigid motions of units smaller than the protein (e.g., secondary structural elements).

The value $\gamma = 6.5$ Å obtained for the LLM fit to staphylococcal nuclease diffuse scattering data is substantially smaller than the 18 Å value obtained by Peck *et al.*[14] for LLM models of diffuse scattering from CypA and WrpA. Peck *et al.*[14] also noted that their values of γ were larger than previously published values, and that the difference in length scales for LLM models might be attributed to their finer sampling of the data. Our results lend support to this explanation for the discrepancy. In the case of intraprotein atom pairs, we found $\gamma = 5.7$ Å, which is comparable to the value $\gamma = 6.5$ Å from the LLM fit. In the case of interprotein atom pairs, we found $\gamma = 14.3$ Å (AMBER) or $\gamma = 13.4$ Å (CHARMM), which is more comparable to the value $\gamma = 18$ Å from the Peck *et al.*[14] study. The similarity of these length scales between the fitting and the MD suggests that both types of motions might be present in the protein crystal. Moreover, the comparison of the length scales between MD analysis and LLM models suggests that the fine-grained sampling might yield data that emphasize interprotein motions, and that the coarse-grained sampling might yield data that emphasize intraprotein motions in the fitting. This possibility motivates future work to identify regions of reciprocal space where the intraprotein signal is enhanced, helping us to realize the vision of connecting diffuse scattering to functionally important motions.

In contrast to the finding by Peck *et al.*[14] that a LLM yielded better agreement with the data than a RBM model of CypA, de-Klijn *et al.*[15] Recently concluded that RBM is the predominant source of diffuse scattering in CypA. Because the connection between the covariance matrix and diffuse scattering is not trivial, it does not logically follow from the results of our covariance analysis that the contribution of RBM to the staphylococcal nuclease diffuse signal is weak. To gain some insight into the importance of RBM in explaining the data, therefore, we fit a simple rigid-body translation model with a single displacement parameter σ to the same data we used to fit the LLM model, enforcing the Laue symmetry (see, e.g., the first term of Eq. (10) in Ref. 16). The refined model had $\sigma = 0.40$ Å, yielding a Pearson correlation coefficient with the anisotropic data of 0.56. The value of σ is almost the same as that for the LLM model, for which $\sigma = 0.41$ Å. The correlation, however, is substantially lower than the value of 0.73 for the LLM model. We therefore conclude that the LLM model more accurately describes the coarsely sampled diffuse scattering data from staphylococcal nuclease. Future work is required to determine why different studies have arrived at different conclusions about the source of diffuse scattering, e.g., the degree to which different data processing and modeling methods are responsible as opposed to differences in what is going on in the crystal different systems.

Ideas from thermal diffuse scattering theory[35] do support the possibility that sampling of diffuse data might preferentially select for different types of motion. As noted by Peck *et al.*,[14] when motions are coupled across unit cell boundaries (as both the present study and their study suggest might be happening in real crystals), the diffuse intensity becomes tied to the Bragg peaks. This effect is closely related to thermal diffuse scattering theory in which the intensity has local maxima at Bragg peak positions and decreases with distance from the peak in a way that is determined by the spectrum of crystal vibrations.[35] Motions coupled on long length scales generate intense features that decay sharply moving away from the peaks, and motions coupled on a shorter length scale contribute less intense features that are spread out over a larger region of reciprocal space, extending farther from the peaks. Because the coarse sampling in our study rejects intensity values within $^1/_4$ of a Miller index of each Bragg peak, it is dominated by the intensity far from the peaks, enriching the signal due to shorter length-scale correlated motions. In contrast, the finer sampling used by Peck *et al.*[14] includes the data corresponding to long length-scale correlated motions, which, despite the localization to fewer grid points near the Bragg peaks, might dominate the fitting due to the higher intensity. In the case of the fine-grained sampling, it is possible that motions on both length scales might be resolved if a LLM with both a short-range and long-range exponential were used.[36]

Although the AMBER and CHARMM simulations yielded similar exponential behavior for the covariance of all Cα atom pairs, a number of differences between the force fields were revealed in our study. (1) The MD simulation pressure after initial solvation was less negative in the case of AMBER (-1439 ± 39 bar) than CHARMM (-1795 ± 252 bar). (2) In both Figs. 3 and 4, the covariance in the case of AMBER stays below zero except at the longest distances, and in the case of CHARMM gradually rises to zero after a minimum at around 55 Å. (3) The short-range behavior differs for the AMBER vs CHARMM simulations. For the CHARMM simulations (Fig. 4), the exponential behavior continues to short distances. In contrast, for the AMBER simulations, the covariance in the 3.31 Å bin is negative: -1.77 ± 1.23 Å^2. (4) The diffuse intensities predicted from the CHARMM simulation had a somewhat higher correlation with the data than those from the AMBER simulation (supplementary Fig. S2). At this time, the origin of the differences is not clear; however, these differences indicate ways in which crystalline MD simulations, including comparisons to diffuse scattering data, have the potential to distinguish between force fields, and therefore might be used to increase force field accuracy.

There are some caveats to consider in interpreting our results. For all but the interprotein atom pairs, the covariance increases sharply below 5 Å, deviating from the values predicted by both the

LLM and RBM models. Such deviation is not surprising, as short-range interactions are more sensitive to the details of the chemical environment, and include interactions between sequential Cα atoms across the rigid peptide bond. The MD simulations were conducted while constraining the distance between all bonded atoms using the linear constraints solver (LINCS) method in GROMACS, which further rigidifies the structure, also tending to increase the covariance. Another caveat is that the exponential fit includes a small negative offset—the correlation function does not decay to zero as the distance increases. (The longest distance corresponds to half the system size, or one lattice vector, along each side of the simulation box.) The offset is nearly zero for the intramolecular case, but is more substantial for the intermolecular and the full supercell analysis, where it is needed to accurately fit the covariance behavior. The offset might be an artifact of the translational alignment of trajectory snapshots,[27] or it might be due to low-frequency crystal vibrations[17] or some other real effect. Note, however, that a constant offset corresponds to a constant long-range covariance, which focuses the diffuse intensity directly beneath the Bragg peaks. In this way, the long-range component of the diffuse intensity might become indistinguishable from the Bragg intensity and not appear in the measured diffuse intensity. A third caveat is that our analysis was performed using only Cα atom pairs, and therefore does not take into account the influence of non-Cα backbone or side chain motions on the covariance behavior. In particular, it is possible that the signature of RBM might not be as clear for side chain atoms as for Cα atoms, if the backbone is more rigid than the side chains. In future work, it will be especially important to analyze simulations in which the bond constraints are relaxed and to overcome the computational difficulties of adding all heavy atoms, including side chain atoms, to the covariance analysis.

Meinhold and Smith[37] performed an analysis of correlated displacements between atom pairs in MD simulations of crystalline staphylococcal nuclease. Instead of analyzing the covariance matrix, however, they analyzed the dependence of elements of the correlation matrix on the distance between atom pairs. The correlation matrix is computed by renormalizing the covariance matrix, dividing each element by the geometric mean of the values on the diagonal (variances) in the corresponding row and column for each element. Meinhold and Smith found that the correlations of all atom pairs decreased exponentially with a decay length of 11 Å, and that interprotein atom pairs also decreased exponentially, with a longer decay length (11–18 Å, depending on the simulation). In this respect, the results of our analyses are similar. However, in contrast to the near linear behavior we found for the covariance of intraprotein atom pairs, they found that the intraprotein correlations showed an exponential decay behavior. Our analysis therefore appears to be inconsistent with theirs with respect to the intraprotein atom pairs. We note that there are several differences between our analyses: on the one hand, Meinhold and Smith[37] analyzed the "correlation" matrix of Cα and other atoms, used a single periodic unit cell, a NPT ensemble, and 10 ns duration simulations; on the other hand, we analyzed the "covariance" matrix of only Cα atoms, used a $2 \times 2 \times 2$ periodic supercell, a NVT ensemble, and 600 ns duration simulations. The difference in the set of atoms used for the analysis seems especially important in light of the expected increased rigidity of Cα atoms compared to all atoms, as discussed in the previous paragraph.

The agreement between the MD simulation and the experimental data differs slightly depending on the details of how the statistics from each 100 ns chunk are accumulated. A strict application of Guinier's equation calls for the statistics to be accumulated coherently, by summing the complex structure factors across each chunk before computing the total diffuse intensity. Here, we also experimented with accumulating the statistics incoherently, by instead averaging the diffuse intensities computed from each 100 ns chunk. Compared to coherent accumulation, incoherent accumulation led to a 0.04–0.05 increase in the Pearson correlation with the experimental data. Although the small size of the difference makes its significance questionable, it is nevertheless worth exploring further, as it suggests that the illuminated volume might more accurately be described as a set of independent domains than as a single crystal. Such a description is consistent with the mosaic block picture, which has been long used to explain crystal imperfections in macromolecular crystallography.[38] It is also consistent with the relatively high concentration of defects that are seen in macromolecular crystals using atomic force microscopy.[39]

As mentioned above, the correlation of the anisotropic intensity calculated from the LLM model with the data is 0.73, which is higher than the maximum value of 0.63 for the MD simulation. The higher correlation of the LLM suggests that it provides a globally more accurate description of the anisotropic intensity. In addition, whereas the LLM is derived from the crystal structure, the MD model drifts away from the crystal structure during the course of the simulation (supplementary Fig. S1). Although the MD model is lacking in this respect, it still has advantages over the LLM model. For one, the MD simulation is still the only model that is capable of reproducing the total intensity—isotropic and anisotropic—and is therefore the most accurate overall. Moreover, the MD model is, in some sense, a "model-free" model, in that there are no free parameters to fit—the model simply depends on the choice of force-field, and the assumption that the system behaves classically. Indeed, the LLM model accuracy is substantially increased due to the ability to refine the free parameters against the experimental data. However accurate the LLM model may be, it produces a very limited description of the dynamics that does not contain any mechanistic information, whereas the MD model can provide us with dynamic structural information that yields functional and biological insight (modulo any inaccuracies inherent to the force field, or due to inadequate sampling and/or simulation length).

Taken together, our results show that MD simulations of a crystalline protein exhibit LLM behavior. The interprotein atom pairs exhibit LLM behavior and within the protein the motions exhibit both LLM and RBM behaviors. Due to the large number of interprotein vs intraprotein atom pairs, the overall behavior appears LLM-like. These findings provide support and context for previous results, which showed that LLM models of protein diffuse scattering improve after the inclusion of interactions across protein boundaries. They also provide clues about why LLM model fits using coarsely sampled diffuse data might yield smaller correlation length scales than using finely sampled data. Finally, our results suggest that the modeling of finely sampled diffuse scattering data might be improved by consideration of both small-scale and large-scale collective motions.

METHODS

Molecular-dynamics simulations

The all atom structure for staphylococcal nuclease was pulled from the Protein Data Bank (wwPDB: 4WOR[40] with unit cell $a = b = 48.5$ Å, $c = 63.43$ Å, $\alpha = \beta = \gamma = 90$ degrees, and space group

P41). This structure is missing the first five residues at the N-terminus and the last eight residues at the C-terminus. In Ref. 28, the missing N- and C-terminal atoms were reintroduced and modeled based on extension of secondary structure—the same starting structure is used in this work.

Once fully modeled, the asymmetric unit was propagated to a unit cell and then to a 2 × 2 × 2 supercell using the "UnitCell" and "PropPDB" methods from AmberTools18.[41] The coordinates of the bound ligand, thymidine-3′-5′-bisphosphate (pdTp), were extracted from the PDB file, saved as a mol2 file (using UCSF Chimera[30]) and parameterized using the SwissParam Server (swissparam.ch[42]). Two different systems were created: one in which the protein residues were parameterized with the AMBER 14SB force field[43] and another in which they were parameterized with the CHARMM 27 force field;[44] both were parameterized using GROMACS[45] "pdb2gmx" (residue names were set manually and hydrogens present in the initial PDB file were ignored with flag-"ignh," which automatically assigns protonation states for residues at pH 7). These fully parameterized systems were then solvated with TIP3P waters[46] using GROMACS "gmx solvate." The full systems were neutralized with chloride ions ("gmx genion"). Once solvated, these systems were minimized using the steepest descent algorithm.

Simulations were performed using a constant particle number, volume, and temperature (NVT) ensemble, at a temperature of 298 K. After an initial round of NVT equilibration to check the pressure of the system, the number of water molecules was adjusted to achieve near-atmospheric pressure. This was achieved by iterative rounds of solvation and NVT equilibration. For the CHARMM force field simulation, 100 ps equilibration durations were used, as in previous studies of the same system.[25,28] For the AMBER force field simulation, 5 ns equilibration durations were used. After the last round of equilibration, 17 557 waters were added in the AMBER simulation, and 17 138 waters were added in the CHARMM simulation.

The crystallographic protein heavy atoms (i.e., nonterminal heavy atoms) were restrained to the minimized crystal structure during all rounds of equilibration and the initial 100 ns restrained production simulation (restraint force constant $k = 1000\,\mathrm{kJ\ mol^{-1}\,nm^{-2}}$). Restraints were then released and production simulation was carried out for 600 ns. All rounds of equilibration and production were carried out using the leap-frog algorithm ("integrator = md"); neighbor searching was carried out using the Verlet cutoff-scheme[47] with an update frequency of 10 frames ("niter = 10"), and a cutoff distance for the short range neighbor list of 1.5 nm ("rlist" = 1.5); all bonds were constrained with the LINCS algorithm ("constraints = all-bonds; constraint-algorithm=LINCS)."[48]

Covariance matrix of atom displacements

After releasing restraints, the simulations require on the order of 100 ns for the RMSD of the protein Cα coordinates to plateau. To ensure that the system was fully equilibrated after the release of restraints, analysis began at 200 ns in to unrestrained production. Cα trajectory subsets were extracted from the full unrestrained production trajectories from 200–600 ns in both simulations. To do this, the first frame (200 ns into unrestrained production) was extracted, and the Cα coordinates were isolated using "gmx editconf" (with flag-pbc to ensure molecules stay whole). Then, a 400 ns Cα trajectory was created using "gmx trjconv," and subsequently translationally fit to the starting structure using the Cα starting frame as reference ("gmx trjconv … -s c_alpha_supercell_pbc.gro … -fit translation)." The Cα trajectory covariance information was calculated using "gmx covar" (once again, using the Cα structure from the first frame as reference).

With 32 proteins, each containing 149 Cα atoms, and a 3 × 3 covariance submatrix for each pair of Cα atoms, the full covariance matrix computed by gmx covar is a 14 304 × 14 304 block matrix. The diagonal elements of this matrix correspond to the mean squared deviation (MSD) for each atom, in each direction; these diagonal elements were ignored in subsequent analysis. After computing the trace of each atoms-pair's 3 × 3 submatrix, the covariance matrix is 4768 × 4768 (supplementary Fig. S3). These pairwise Cα covariances were sorted by their distances apart, using a matrix of pairwise Cα distances (supplementary Fig. S3) computed using MDTraj[49] and the average coordinates reported by gmx covar for each Cα trajectory subset. Covariance matrix data were processed, analyzed, and plotted using python's "numpy, scipy, and matplotlib." The covariance as a function of distance data was fit to an exponential decay model using "gnuplot," and errors in the parameters reported in the Results section are the asymptotic standard errors reported by gnuplot.

Rigid-body motions model

To investigate the source of the nonexponential decrease in covariance as a function of distance for residue pairs within proteins, we created a Python script to simulate RBM ("RigidBodyMotions.py" in the supplementary material and at https://github.com/mewall/lunus/blob/master/scripts/RigidBodyMotions.py). The script creates hypothetical trajectories consisting of a rigid protein randomly rotated and translated by amounts on a par with the magnitude of motion observed in our MD simulations. We compared the covariance behavior computed from these "trajectories" with that observed in the crystalline MD simulations. This allowed us to determine the degree to which a rigid-body rotation and translation model can explain the MD covariance behavior.

The Python script generates covariance data as follows:

(1) The structure of a single protein is pulled from the crystalline MD starting structure, and centered on the origin "(using mdtraj.Trajectory().center_coordinates());" terminal atoms missing from the PDB structure are disregarded, as they cannot reasonably be considered rigid.
(2) Rotations are generated by sampling three Euler angles, each from a normal distribution with mean zero and a specified standard deviation; similarly, translations are generated by sampling a three-dimensional vector from a normal distribution with mean at the origin and a specified standard deviation, the same for all directions.
(3) A new "frame" is created by first rotationally moving and then translationally moving the starting structure; for rotational moves, a rotation matrix is generated from the Euler angles, and the matrix vector product of the rotation matrix and the coordinates of each atom is performed (with "numpy.dot());" for translations, the random three-dimensional translation vector is added to each atom's coordinates.
(4) A "trajectory" is built up frame by frame, and the covariance is calculated as a function of the distance between atoms, as described above for simulation trajectories.

In this study, we used 5000 frames for our analysis. The covariance data produced were compared with the covariance data from equivalent atoms in the simulation (nonterminal Cα atom pairs within proteins). Reasonable parameters for the Euler angle and translational distributions were arrived at by manual adjustment of the standard deviations, seeking the best visual overlap between the model and the data.

Diffuse scattering

Diffuse scattering data for staphylococcal nuclease were obtained from past experiments[40] and were processed as described in Ref. 28. In addition to studying the LLM behavior of the atom displacement covariance matrix, the MD trajectories themselves can be processed directly to predict the diffuse scattering. The diffuse scattering is the variance of the structure factor of independent repeating units in the crystal, according to Guinier's equation,[50] and previous studies have predicted the diffuse scattering from protein crystal MD trajectories by computing the structure factor, frame-by-frame.[25,27,28,37,51] This is done as in Ref. 28 using the script "get_diffuse_from_md.py," which takes in a trajectory and outputs .mtz (byte stream) and *.dat* (ascii) reflection data; then, reflected data are processed with the "Lunus" diffuse scattering data processing software suite (https://github.com/mewall/lunus). The same script and processing software were used in this work.

Diffuse scattering simulations were calculated from the last 400 ns of the trajectories. As in Ref. 28, the intensities were computed on a 3D grid sampling reciprocal space twice as finely as the Bragg lattice. The trajectory was produced in 100 ns chunks, and the diffuse scattering was calculated from these 100 ns chunks independently, and then accumulated either (a) "coherently," by accumulating the complex structure factors ("flag-merge = True") or (b) incoherently, by averaging the intensities themselves from each 100 ns chunk using Lunus methods "sumlt" and "mulsclt." The model diffuse scattering was converted to a lattice file using "hkl2lat," with the experimental lattice file as a template, then symmetrized, and culled by resolution range using "symlt" and "culllt," and the anisotropic component of the diffuse scattering was computed using "anisolt." Pearson correlations between the models and the data were computed using "corrlt."

Simulated diffraction images in Fig. 2 and supplementary Fig. S4 were computed from models or data in a similar way to that in Ref. 28. To simulate diffraction using a 3D grid model or data, the minimum value was computed within shells and was subtracted from each 3D grid point, using interpolation ("subminlt"). The diffraction pattern corresponding to a specified crystal orientation was simulated from the 3D grid using the "simulate_diffraction_image.py" Python script distributed with Lunus.

For the refinement of the LLM, we processed the data using a recent version of Lunus (https://github.com/mewall/lunus). As in the original staphylococcal nuclease study,[40] the data were coarsely sampled using one point per Miller index. The data were processed to a resolution limit of 1.6 Å. Intensity values within 1/4 of a Miller index of each Bragg peak were excluded from the processing. The 3D data were symmetrized using the P4 Laue symmetry, and the isotropic component was removed as described in Ref. 28. The structure factors from the 4WOR crystal structure were used to refine a LLM model, using the refine_llm.py script distributed with Lunus ("python refine_llm.py symop=-3 model=llm bfacs=zero)."

SUPPLEMENTARY MATERIAL

See the supplementary material for five supplementary figures and a Python script: RMSD of the supercell C-alpha coordinates between the crystal structure and the MD model (FIG. S1); agreement between 3D diffuse scattering data and simulations computed using sections of the MD trajectory (FIG. S2); visualization of values from the Cα atom displacement covariance matrix and the Cα atom distance matrix (FIG. S3); simulated diffraction images from the LLM model and 3D experimental diffuse data (FIG. S4); number of Cα atom pairs as a function of distance in the full supercell (FIG. S5); and RigidBodyMotions.py, a Python script used to compute the covariance matrix of Cα displacements in a model of rigid-body rotations and translations and to compute the mean covariance in bins of atom pair distance. The script is also available at https://github.com/mewall/lunus/blob/master/scripts/RigidBodyMotions.py.

ACKNOWLEDGMENTS

This work was supported by the University of California Laboratory Fees Research Program (No. LFR-17-476732). M.E.W. was also supported by the Exascale Computing Project (No. 17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. J.S.F. was also supported by NSF No. STC-1231306. The simulations were performed using Institutional Computing machines at the Los Alamos National Laboratory, supported by the U.S. Department of Energy under Contract No. 89233218CNA000001.

REFERENCES

[1]D. A. Keen and A. L. Goodwin, Nature **521**(7552), 303–309 (2015).
[2]M. E. Wall, A. M. Wolff, and J. S. Fraser, Curr. Opin. Struct. Biol. **50**, 109–116 (2018).
[3]S. P. Meisburger, W. C. Thomas, M. B. Watkins, and N. Ando, Chem. Rev. **117**(12), 7615–7672 (2017).
[4]J. M. Holton, S. Classen, K. A. Frankel, and J. A. Tainer, FEBS J. **281**(18), 4046–4060 (2014).
[5]H. van den Bedem and J. S. Fraser, Nat. Methods **12**(4), 307–318 (2015).
[6]V. Schomaker and K. N. Trueblood, Acta Crystallogr., Sect. B **24**(1), 63–76 (1968).
[7]F. Zucker, P. C. Champ, and E. A. Merritt, Acta Crystallogr., Sect. D **66**(Pt 8), 889–900 (2010).
[8]R. A. Woldeyes, D. A. Sivak, and J. S. Fraser, Curr. Opin. Struct. Biol. **28**, 56–62 (2014).
[9]D. A. Keedy, J. S. Fraser, and H. van den Bedem, PLoS Comput. Biol. **11**(10), e1004507 (2015).
[10]J. L. Smith, W. A. Hendrickson, R. B. Honzatko, and S. Sheriff, Biochemistry **25**(18), 5018–5027 (1986).
[11]P. B. Moore, Structure **17**(10), 1307–1315 (2009).
[12]A. H. Van Benschoten, P. V. Afonine, T. C. Terwilliger, M. E. Wall, C. J. Jackson, N. K. Sauter, P. D. Adams, A. Urzhumtsev, and J. S. Fraser, Acta Crystallogr., Sect. D **71**(Pt 8), 1657–1667 (2015).
[13]M. E. Wall, P. D. Adams, J. S. Fraser, and N. K. Sauter, Structure **22**(2), 182–184 (2014).
[14]A. Peck, F. Poitevin, and T. J. Lane, IUCrJ **5**(Pt 2), 211–222 (2018).
[15]T. de Klijn, A. M. M. Schreurs, and L. M. J. Kroon-Batenburg, IUCrJ **6**(Pt 2), 277–289 (2019).
[16]H. N. Chapman, O. M. Yefanov, K. Ayyer, T. A. White, A. Barty, A. Morgan, V. Mariani, D. Oberthuer, and K. Pande, J. Appl. Crystallogr. **50**(Pt 4), 1084–1103 (2017).
[17]Y. S. Polikanov and P. B. Moore, Acta Crystallogr., Sect. D **71**(Pt 10), 2021–2031 (2015).

[18]D. L. Caspar, J. Clarage, D. M. Salunke, and M. Clarage, Nature **332**(6165), 659–662 (1988).
[19]B. T. Burnley, P. V. Afonine, P. D. Adams, and P. Gros, eLife **1**, e00311 (2012).
[20]E. J. Levin, D. A. Kondrashov, G. E. Wesenberg, and G. N. Phillips, Jr., Structure **15**(9), 1040–1052 (2007).
[21]D. Riccardi, Q. Cui, and G. N. Phillips, Jr., Biophys. J. **99**(8), 2616–2625 (2010).
[22]W. F. van Gunsteren and M. Karplus, Nature **293**(5834), 677–678 (1981).
[23]P. A. Janowski, D. S. Cerutti, J. Holton, and D. A. Case, J. Am. Chem. Soc. **135**(21), 7938–7948 (2013).
[24]P. A. Janowski, C. Liu, J. Deckman, and D. A. Case, Protein Sci. **25**(1), 87–102 (2016).
[25]M. E. Wall, A. H. Van Benschoten, N. K. Sauter, P. D. Adams, J. S. Fraser, and T. C. Terwilliger, Proc. Natl. Acad. Sci. U. S. A. **111**(50), 17887–17892 (2014).
[26]J. B. Clarage, T. Romo, B. K. Andrews, B. M. Pettitt, and G. N. Phillips, Jr., Proc. Natl. Acad. Sci. U. S. A. **92**(8), 3288–3292 (1995).
[27]L. Meinhold and J. C. Smith, Biophys. J. **88**(4), 2554–2563 (2005).
[28]M. E. Wall, IUCrJ **5**(Pt 2), 172–181 (2018).
[29]M. E. Wall, IUCrJ **5**(Pt 2), 120–121 (2018).
[30]E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, J. Comput. Chem. **25**(13), 1605–1612 (2004).
[31]A. Arvai, *ADXV—A Program to Display X-Ray Diffraction Images* (Scripps Research Institute, La Jolla, CA, 2012).
[32]V. Kurauskas, S. A. Izmailov, O. N. Rogacheva, A. Hessel, I. Ayala, J. Woodhouse, A. Shilova, Y. Xue, T. Yuwen, N. Coquelle, J. P. Colletier, N. R. Skrynnikov, and P. Schanda, Nat. Commun. **8**(1), 145 (2017).
[33]P. Ma, Y. Xue, N. Coquelle, J. D. Haller, T. Yuwen, I. Ayala, O. Mikhailovskii, D. Willbold, J. P. Colletier, N. R. Skrynnikov, and P. Schanda, Nat. Commun. **6**, 8361 (2015).
[34]J. Doucet and J. P. Benoit, Nature **325**(6105), 643–646 (1987).
[35]R. W. James, *The Optical Principles of the Diffraction of X-Rays* (G. Bell and Sons, London, 1948).
[36]J. B. Clarage, M. S. Clarage, W. C. Phillips, R. M. Sweet, and D. L. Caspar, Proteins **12**(2), 145–157 (1992).
[37]L. Meinhold and J. C. Smith, Proteins **66**(4), 941–953 (2007).
[38]J. R. Helliwell, J. Cryst. Growth **90**(1–3), 259–272 (1988).
[39]A. J. Malkin, Y. G. Kuznetsov, and A. McPherson, J. Struct. Biol. **117**(2), 124–137 (1996).
[40]M. E. Wall, S. E. Ealick, and S. M. Gruner, Proc. Natl. Acad. Sci. U. S. A. **94**(12), 6180–6184 (1997).
[41]D. A. Case, I. Y. Ben-Shalom, S. R. Brozell, D. S. Cerutti, I. T. E. Cheatham, V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, D. Ghoreishi, M. K. Gilson, H. Gohlke, A. W. Goetz, D. Greene, R. Harris, N. Homeyer, S. Izadi, A. Kovalenko, T. Kurtzman, T. S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D. J. Mermelstein, K. M. Merz, Y. Miao, G. Monard, C. Nguyen, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R. Qi, D. R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C. L. Simmerling, J. Smith, R. Salomon-Ferrer, J. Swails, R. C. Walker, J. Wang, H. Wei, R. M. Wolf, X. Wu, L. Xiao, D. M. York, and P. A. Kollman, *AMBER 2018* (University of California, San Francisco, 2019).
[42]V. Zoete, M. A. Cuendet, A. Grosdidier, and O. Michielin, J. Comput. Chem. **32**(11), 2359–2368 (2011).
[43]J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, and C. Simmerling, J. Chem. Theory Comput. **11**(8), 3696–3713 (2015).
[44]A. D. MacKerell, Jr., N. Banavali, and N. Foloppe, Biopolymers **56**(4), 257–265 (2000).
[45]H. J. C. Berendsen, D. van der Spoel, and R. van Drunen, Comput. Phys. Commun. **91**(1), 43–56 (1995).
[46]W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, J. Chem. Phys. **79**(2), 926–935 (1983).
[47]S. Páll and B. Hess, Comput. Phys. Commun. **184**(12), 2641–2650 (2013).
[48]B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. Fraaije, J. Comput. Chem. **18**(12), 1463–1472 (1997).
[49]R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, and V. S. Pande, Biophys. J. **109**(8), 1528–1532 (2015).
[50]A. Guinier, *X-Ray Diffraction in Crystals, Imperfect Crystals, and Amorphous Bodies* (W. H. Freeman and Company, San Francisco, 1963).
[51]L. Meinhold and J. C. Smith, Phys. Rev. Lett. **95**(21), 218103 (2005).

Supplementary Information

Liquid-like and rigid-body motions in molecular-dynamics simulations of a crystalline protein

David C. Wych[1,2], James S. Fraser[3], David L. Mobley[1,4], Michael E. Wall[2,*]

[1]Department of Pharmaceutical Sciences, University of California, Irvine, Irvine, CA, 92697, USA
[2]Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, 87545, USA
[3]Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, San Francisco, CA, 94143, USA
[4]Department of Chemistry, University of California, Irvine, Irvine, CA, 92697, USA

Los Alamos National Laboratory LA-UR-19-30460

Abbreviations: molecular dynamics (MD); liquid-like motions (LLM); rigid-body motions (RBM); thymidine-3′-5′-bisphosphate (pdTp)

Key words: liquid-like motions; rigid-body motions; molecular-dynamics simulation; protein crystallography; protein dynamics; diffuse scattering

*Corresponding author: mewall@lanl.gov

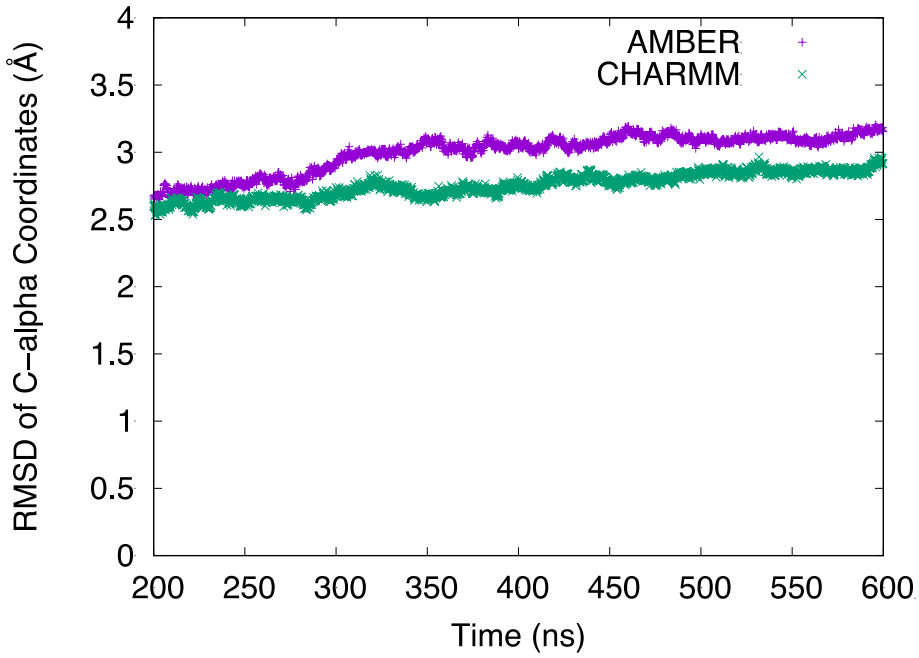

Figure S1. RMSD of the supercell C-alpha coordinates between the crystal structure and the MD model. Values were sampled every 200 ps. AMBER simulation values are shown as purple '+' symbols and CHARMM values using green '×' symbols.

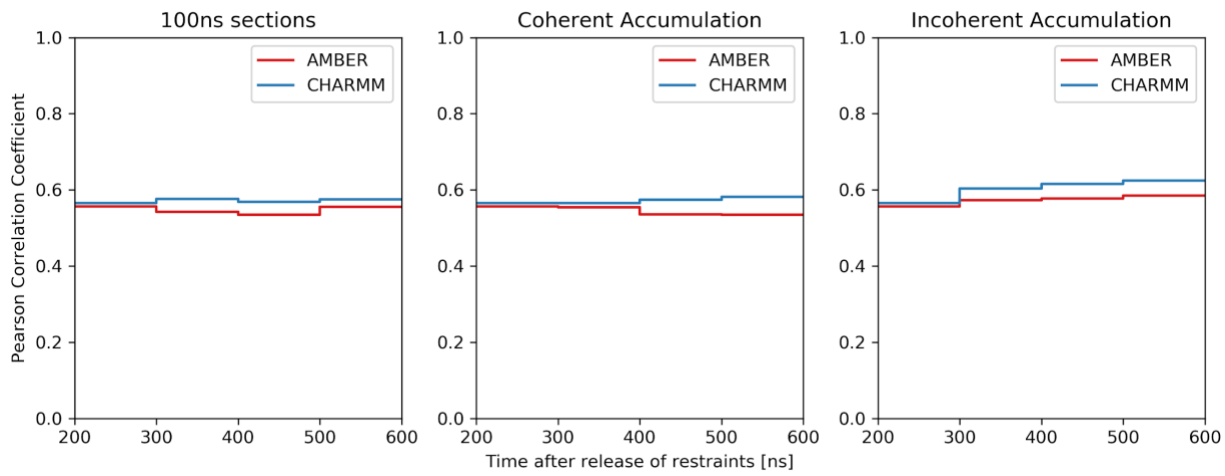

Figure S2. Agreement between 3D diffuse scattering data and simulations computed using sections of the MD trajectory. Values of the Pearson correlation coefficient are computed as a figure of merit. AMBER values are in red, CHARMM in blue. *Left.* Agreement computed using individual 100 ns sections after the 200 ns time point (200-300, 300-400, 400-500, and 500-600 ns). Values do not systematically increase or decrease with the simulation time. *Center*. Simulations computed using trajectories accumulated coherently after the 200 ns time point (see Methods for definition of coherent accumulation): (200-300, 200-400, 200-500, and 200-600 ns). The AMBER values decrease and the CHARMM values increase with increasing time. *Right.* Simulations computed using trajectories accumulated incoherently across 100 ns sections after the 200 ns time point (see Methods for definition of incoherent accumulation). Both the AMBER and CHARMM values increase with increasing time. Incoherent accumulation results in the highest correlation with the diffuse scattering data, and the CHARMM correlation is higher than the AMBER correlation in all cases.

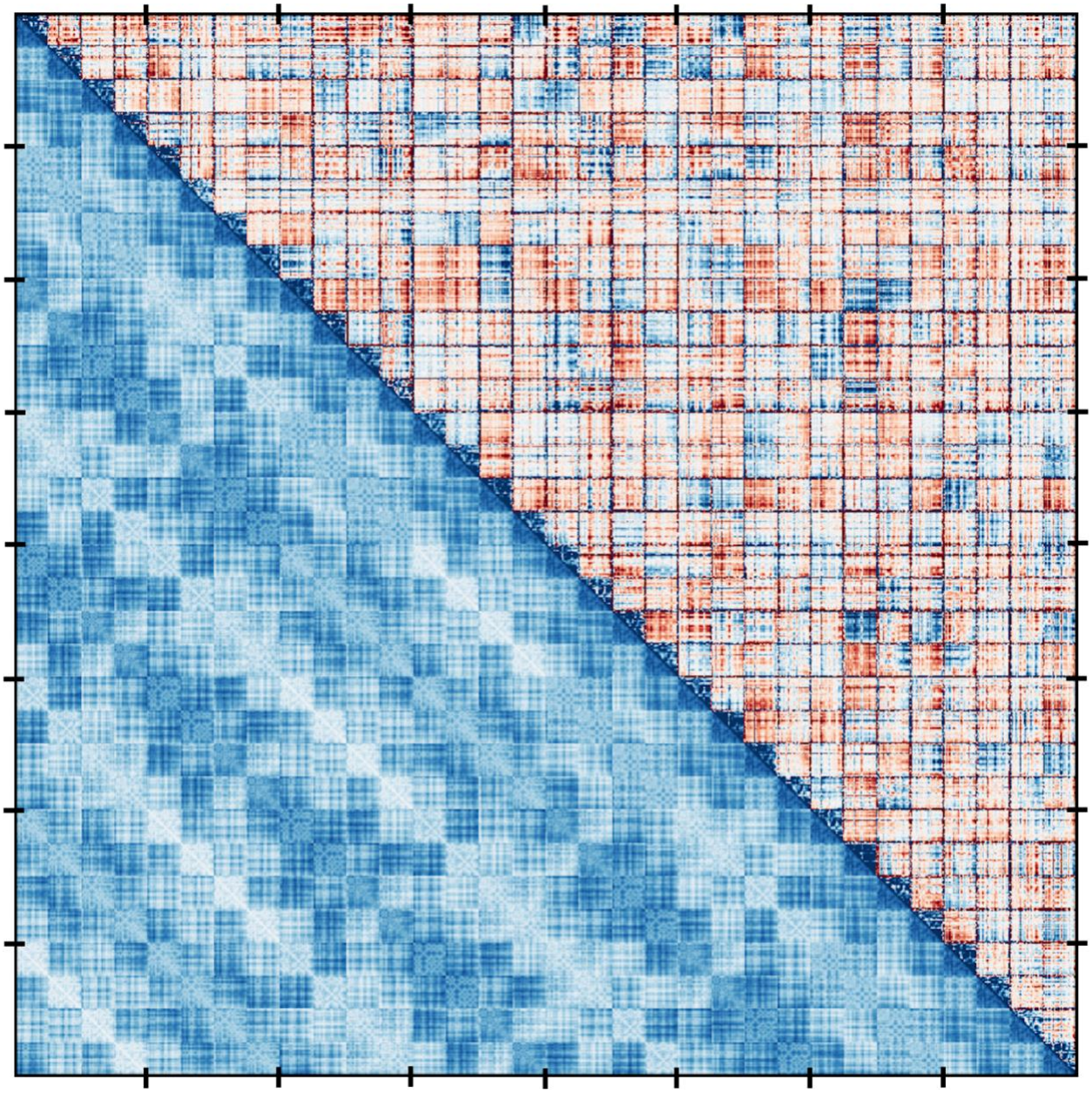

Figure S3. Visualization of values from the Cα atom displacement covariance matrix (*upper right triangle*) and the Cα atom distance matrix (*lower left triangle*). Rows and columns correspond to Cα atoms (the image shows only a sampling of the values from the full matrix and merely illustrates the overall structure of the data, as the resolution is insufficient to show values for all atom pairs). Tics indicate breaks between each of eight unit cells. The finer squares within each unit cell correspond to four copies of the proteins, each of which has 149 Cα atoms. The covariance values in the upper right triangle range from -0.2 $Å_2$ (*dark red*) to 0 (*white*) to + 0.2 $Å_2$ (*dark blue*), with values above or below in absolute value visualized as the same dark blue/red color. The distance values in the lower left triangle range from 0 (*dark blue*) to 92 Å (*white*). The blocks along the diagonal correspond to intra-protein atom pairs; for these blocks, the covariance tends to be positive (*blue*) and the distances are short (also *blue*). All other blocks correspond to inter-protein atom pairs; for these blocks, there are more negative covariance values than for the intra-protein blocks.

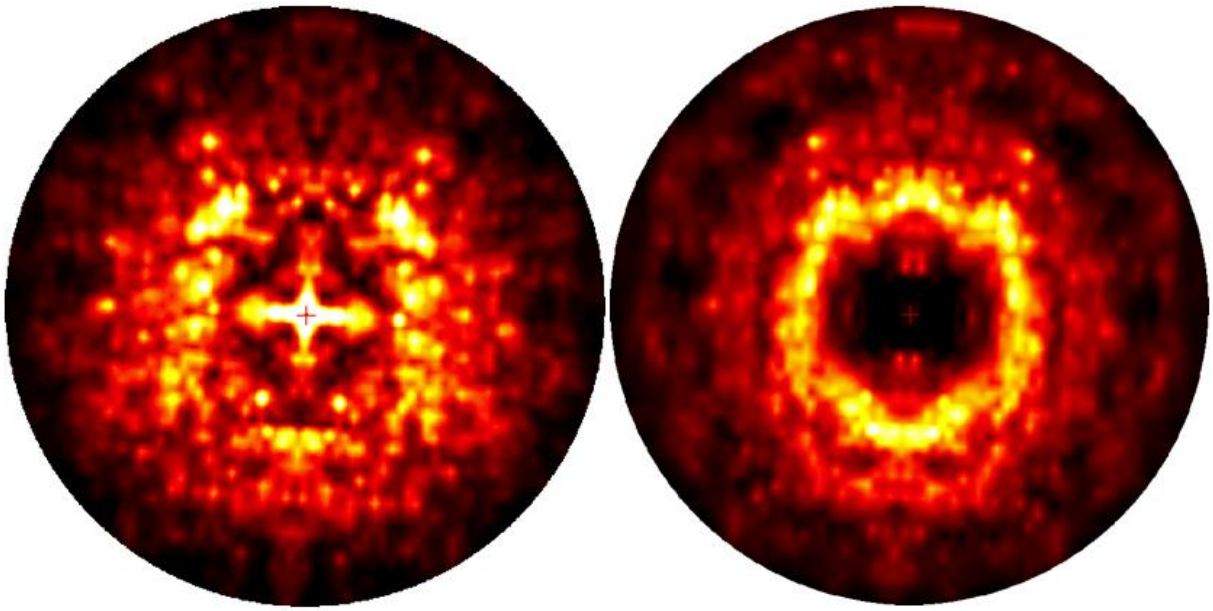

Figure S4. Simulated diffraction images from LLM model (*left*) and 3D experimental diffuse data (*right*). Both the data and model were generated using coarse sampling of one measurement per Miller index. Outside of the region at low resolution near the origin (which, despite the small contribution of this region in determining the quantitative agreement, can dominate the visual comparison due to its central position), similar features are identifiable between the two images. The images were displayed using the heat map mode of ADXV.[31]

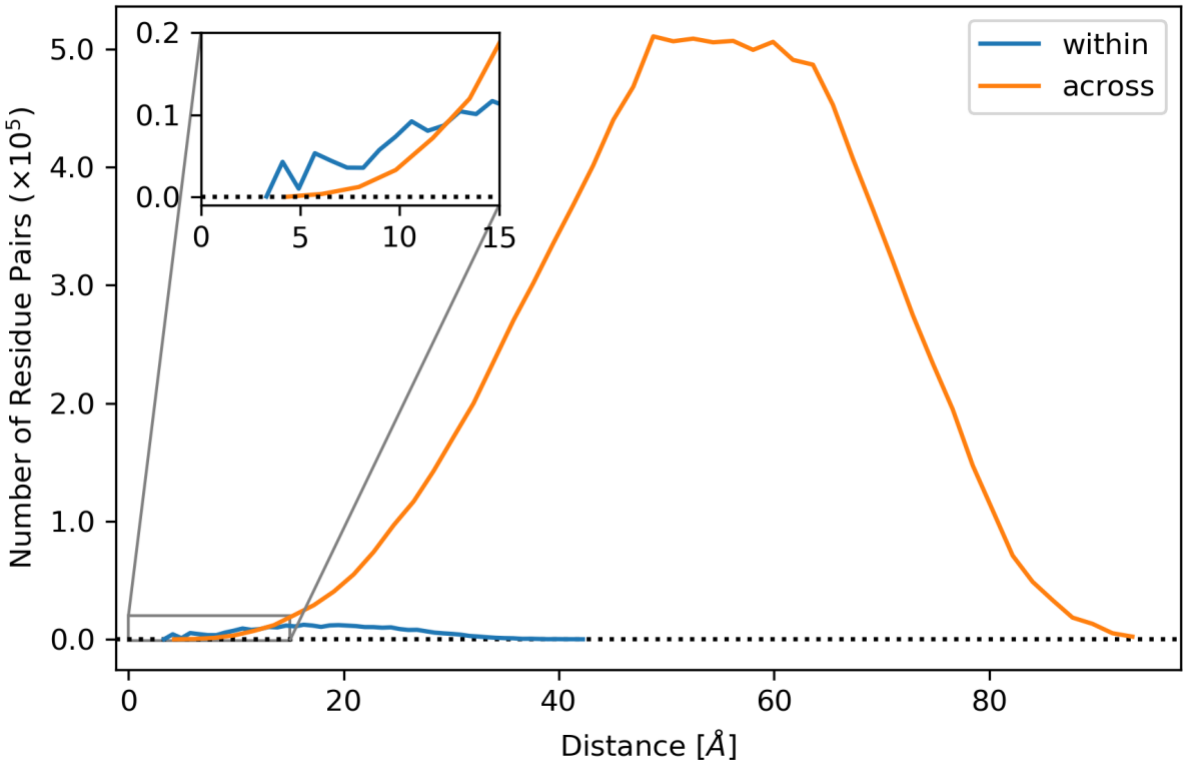

Figure S5. Number of Cα atom pairs as a function of distance in the full supercell. Intra-protein (within) atom pairs are shown in blue; inter-protein (across) atom pairs are shown in orange. The number of pairs across proteins outnumber the residue pairs within proteins for all distances greater than 12 Å (*inset*).

```
# RigidBodyMotions.py script
# Used for covariance analysis in DW Wych, JS Fraser, DL Mobley, ME Wall. Liquid-like and
#     rigid-body motions in molecular-dynamics simulations of a crystalline protein.
#     LA-UR-19-30460
# Author: David Wych, UC Irvine and Los Alamos National Laboratory
# Date: August 2019

import mdtraj as md
import numpy as np
from scipy.stats import sem

class Model():
    """Rigid Body Motions model for protein c-alpha
    residue pair covariance as a function of distance"""
    def rotation(self):
        def eulerAnglesToRotationMatrix(eulers):
            """ function for converting vector of 3 euler angles
            to a rotation matrix"""
            R_x = np.array([[1,0,0],
                            [0,np.cos(eulers[0]),-np.sin(eulers[0])],
                            [0,np.sin(eulers[0]),np.cos(eulers[0])]
                            ])
            R_y = np.array([[np.cos(eulers[1]),0,np.sin(eulers[1])],
                            [0,1,0],
                            [-np.sin(eulers[1]),0,np.cos(eulers[1])]
                            ])
            R_z = np.array([[np.cos(eulers[2]),-np.sin(eulers[2]),0],
                            [np.sin(eulers[2]),np.cos(eulers[2]),0],
                            [0,0,1]
                            ])
            R = np.dot(R_z, np.dot( R_y, R_x ))
            return R

        # For every frame but the first...
        for t in range(len(self.traj)-1):
            # Generate a new vector of three euler angles and convert
            # it to a matrix
            _eulers = np.deg2rad(np.random.normal(0.0, self.rot_std, 3))
            R_mat = eulerAnglesToRotationMatrix(_eulers)
            # Apply that matrix to the coordiantes of every atom in each frame
            for n in range(self.traj.shape[1]):
                self.traj[t+1][n] = np.dot(R_mat, self.traj[t+1][n])

    def translation(self):
        # For every frame but the first...
        for t in range(len(self.traj)-1):
```

```
            # Generate a random 3-d translation vector
            _trans = np.random.normal(0, self.trans_std, 3)
            # Add that translation to each atom in the
            self.traj[t+1] = np.add(self.traj[t+1], _trans)

    def calc_covarmat(self):
        X = self.traj[:,:,0]; Y = self.traj[:,:,1]; Z = self.traj[:,:,2]
        uX = np.mean(X, axis=0); uY = np.mean(Y, axis=0); uZ = np.mean(Z, axis=0)
        covXX = np.mean(np.array([np.outer(np.subtract(vec, uX), np.subtract(vec, uX)) for  vec in
X]), axis=0)
        covYY = np.mean(np.array([np.outer(np.subtract(vec, uY), np.subtract(vec, uY)) for  vec in
Y]), axis=0)
        covZZ = np.mean(np.array([np.outer(np.subtract(vec, uZ), np.subtract(vec, uZ)) for  vec in
Z]), axis=0)
        covar = np.add(covXX, np.add(covYY, covZZ))
        self.covarmat = covar

    def calc_distances(self):
        N = self.prot.xyz.shape[1]
        xmat = np.tile(self.prot.xyz[0][:,0], N).reshape((N,N))
        ymat = np.tile(self.prot.xyz[0][:,1], N).reshape((N,N))
        zmat = np.tile(self.prot.xyz[0][:,2], N).reshape((N,N))

        xdiff = xmat.T - xmat
        ydiff = ymat.T - ymat
        zdiff = zmat.T - zmat

        total_diff = np.sqrt(np.power(xdiff, 2) + np.power(ydiff, 2) + np.power(zdiff, 2))

        self.distmat = 10.0*total_diff

    def collect_bins(self, n_bins=35):
        udi = np.triu_indices(self.covarmat.shape[0], 1)
        covar_scatter = self.covarmat[udi].flatten()
        dist_scatter = self.distmat[udi].flatten()
        # Sort the distance and covariance arrays
        dist_idx = np.argsort(dist_scatter)
        dist_sorted = dist_scatter[dist_idx]
        covar_sorted = covar_scatter[dist_idx]
        # Create array of bins width "bin_width"
        # with range "range - bin_width"
        #dist_bins = np.linspace(0, np.max(dist_scatter), num=n_bins)
        dist_bins = self.dist
        # Create array of bins for covariance with
        # the same number of bins as "dist_bins"
        covar_mean_bins = []
```

```
        covar_sem_bins = []
        for i in range(len(dist_bins)-1):
            _dist_last = dist_bins[i]
            _dist = dist_bins[i+1]
            _dist_idx_1 = np.where(dist_sorted >= _dist_last)
            _dist_idx_2 = np.where(dist_sorted <= _dist)
            _dist_idx = np.intersect1d(_dist_idx_1, _dist_idx_2)
            _dist_subarray = dist_sorted[_dist_idx]
            _covar_subarray = covar_sorted[_dist_idx]
            covar_mean_bins.append(np.mean(_covar_subarray))
            covar_sem_bins.append(sem(_covar_subarray))

        covar_mean_bins = np.array(covar_mean_bins)
        covar_sem_bins = np.array(covar_sem_bins)

        return dist_bins[1:], covar_mean_bins, covar_sem_bins

    def __init__(self, dist=None, MODEL=None, N_FRAMES=None, ROT_STD=None,
TRANS_STD=None):
        if type(dist) != np.ndarray:
            self.dist = np.array([3.30216069+(0.8112175*(i-1)) for i in range(50)])
        else:
            self.dist = dist
            self.dist = np.append([self.dist[0]-(self.dist[1] - self.dist[0])], list(self.dist))
        if MODEL == None:
            self.model = "rot+trans"
        else:
            self.model = MODEL
        if N_FRAMES == None:
            self.n_frames = 100
        else:
            self.n_frames = N_FRAMES
        if ROT_STD == None:
            self.rot_std = 1.25
        else:
            self.rot_std = ROT_STD
        if TRANS_STD == None:
            self.trans_std = 0.025
        else:
            self.trans_std = TRANS_STD


        sys = md.load("4wor_complete.pdb")
        top = sys.topology
        self.prot = sys.atom_slice(top.select("name 'CA' and (resid 6 to 141)"))
        self.prot.center_coordinates()
```

```
    self.coords = self.prot.xyz
    self.traj = np.array(list(self.coords)*self.n_frames)

def run(self):
    if self.model == "rot":
        self.rotation()
        self.calc_covarmat()
        self.calc_distances()
        dist_bins_model, covar_means_model, covar_sems_model = self.collect_bins()
        return dist_bins_model, covar_means_model, covar_sems_model
    if self.model == "trans":
        self.translation()
        self.calc_covarmat()
        self.calc_distances()
        dist_bins_model, covar_means_model, covar_sems_model = self.collect_bins()
        return dist_bins_model, covar_means_model, covar_sems_model
    if self.model == "rot+trans":
        self.rotation()
        self.translation()
        self.calc_covarmat()
        self.calc_distances()
        dist_bins_model, covar_means_model, covar_sems_model = self.collect_bins()
        return dist_bins_model, covar_means_model, covar_sems_model
```