Open Science Grid Study of the Coupling between Conformation and Water Content in the Interior of a Protein

Ana Damjanović,*[,†,‡] Benjamin T. Miller,[‡] Torre J. Wenaus,[§] Petar Maksimović,[||] Bertrand García-Moreno E.,[†] and Bernard R. Brooks[‡]

Department of Biophysics, Johns Hopkins University, Baltimore, Maryland, Laboratory of Computational Biology, National Heart, Lung and Blood Institute, National Institutes of Health, Bethesda, Maryland, Physics Department, Brookhaven National Laboratory, Upton, New York, and Department of Physics, Johns Hopkins University, Baltimore, Maryland

Received August 7, 2008

Computational grids are a promising resource for modeling complex biochemical processes such as protein folding, penetration of gases or water into proteins, or protein structural rearrangements coupled to ligand binding. We have enabled the molecular dynamics program CHARMM to run on the Open Science Grid. The implementation is general, flexible, easily modifiable for use with other molecular dynamics programs and other grids and automated in terms of job submission, monitoring, and resubmission. The usefulness of grid computing was demonstrated through the study of hydration of the Glu-66 side chain in the interior of protein staphylococcal nuclease. Multiple simulations started with and without two internal water molecules shown crystallographically to be associated with the side chain of Glu-66 yielded two distinct populations of rotameric states of Glu-66 that differed by as much as 20%. This illustrates how internal water molecules can bias protein conformations. Furthermore, there appeared to be a temporal correlation between dehydration of the side chain and conformational transitions of Glu-66. This example demonstrated how difficult it is to get convergence even in the relatively simple case of a side chain oscillating between two conformations. With grid computing, we also benchmarked the self-guided Langevin dynamics method against the Langevin dynamics method traditionally used for temperature control in molecular dynamics simulations and showed that the two methods yield comparable results.

INTRODUCTION

Molecular dynamics (MD) simulations, the primary tool for computational modeling of biomolecular dynamics, can rarely sample the full spectrum of conformational states accessible to biomolecules. The large number of atoms that are needed to describe a protein or a DNA molecule embedded in a solvent or lipid environment limits the timescales achievable by routine MD simulations to tens and hundreds of nanoseconds. Only a few microsecond MD simulations have been reported for proteins or other larger systems.[1] For a large number of biologically important motions, such as side-chain rotations, penetration of water or diffusion of gases in proteins, the sampling of pathways for such processes, or gathering of meaningful statistics needed for determination of experimentally measurable quantities, is often insufficient, even though relevant events are observed during the simulations.[2]

Important biochemical processes such as protein folding or conformational rearrangements usually occur on timescales longer than microseconds. The probability of observing a conformational transition in a single MD simulation is extremely small. Computational methods that go beyond classical MD simulations are needed. Such methods are based on different types of algorithms,[3] e.g., coarse graining, smoothing protocols, generalized ensembles, optimization of actions, or additions of constraints or forces. Some methods, such as replica-exchange molecular dynamics (REXMD), require simultaneous simulations on multiple processors.[4,5] Even using methods that do not have such requirements, such as self-guided Langevin dynamics (SGLD),[6] the desired transitions may not be observable in a single simulation,[2] and multiple simulations, preferably on hundreds and thousands of processors, may be needed to observe even a single event.

New computational methods have to be extensively benchmarked against more traditional tools. Furthermore, there is a pressing need for comparison of force-fields used in MD simulations. Due to the stochastic nature of MD simulations, the different methods and force-fields can only be compared when statistically meaningful samples, such as those provided by multiple simulations, are obtained.

Distributed computing is a promising resource for performing the multiple MD simulations needed for statistically meaningful modeling of biological processes. Some applications that employ distributed computing to biological problems such as protein folding and structure prediction are Folding@home,[7] Predictor@home,[8] and Rosetta@home.[9] These projects take advantage of unused cycles of computational resources around the world, especially personal computers belonging to individuals interested in donating their computer time to scientific research. Computational

* Corresponding author. E-mail: ad@jhu.edu.
[†] Department of Biophysics, Johns Hopkins University.
[‡] National Heart, Lung and Blood Institute.
[§] Brookhaven National Laboratory.
[||] Department of Physics, Johns Hopkins University.

grids can also be composed of institutionally owned and operated computer systems that are linked together by a common software stack to present a uniform method of resource access. This stack generally includes mechanisms for user authentication, job submission, and research monitoring.[10]

Foreseeing the utility of linking together various high performance computing centers, projects such as the Open Science Grid (OSG),[11] the TeraGrid,[12] and the EUROGRID[13] were established to bring together computing and storage resources into grid infrastructure. For example, the computational resources for the OSG come from universities, national laboratories, and computing centers across the United States, Brazil, Taiwan, and the UK. Unfortunately, these projects do not offer an easy way for scientists to run their scientific software on the grid.

Several software tools have recently been developed to utilize grid computing for biological applications.[14−17] Grid enabled versions of major molecular dynamics programs such as CHARMM,[18] NAMD,[19] AMBER,[20] and GROMACS[21,22] have been developed.

Previous methods for running CHARMM[23] on a computational grid have focused primarily on the mechanics of running the program in a distributed environment and on proving the utility of using CHARMM with such systems. The work presented here expands on this by allowing researchers to use grid resources in a more flexible and decoupled fashion. This goal was accomplished by providing software, the so-called grid executor, with the following characteristics: (i) It is **automated**—the grid executor submits, monitors, and resubmits the jobs, such that the amount of time a researcher spends in preparation and babysitting of jobs is minimized. (ii) It is **general**—it allows a user to distribute the workload over many different systems and sites without needing to know details about how the sites are set up. It also eliminates the need for the extra portal software that many previously published grid programs require. (iii) It is **flexible**—the user specifies the types of workflows that are to be executed. It can be used with any molecular dynamics program, not just CHARMM, and in principle, it is not limited to the OSG.

Achieving this was challenging because of issues with different hardware environments at the remote sites as well as with data locality and program control. A control mechanism had to be devised that would allow a researcher without direct access to grid storage and staging resources to describe the work that needs to be performed, define where the output data should be stored, and have that work be completed with minimal further human interaction. Finally, computing resources on the OSG amenable to running CHARMM in the desired manner needed to be identified, and experimentation was required to determine the appropriate run time for the individual jobs. Because the optimal queues for running CHARMM on the grid ended up having a 12 h limit, all MD runs needed to be additionally split into smaller runs. With such a large number of simulations that needed to be managed, automatic submission, monitoring, and resubmission featured by the grid executor became indispensable, not only to minimizes the possibility of errors, but also to save the time that the researcher spends performing the simulations.

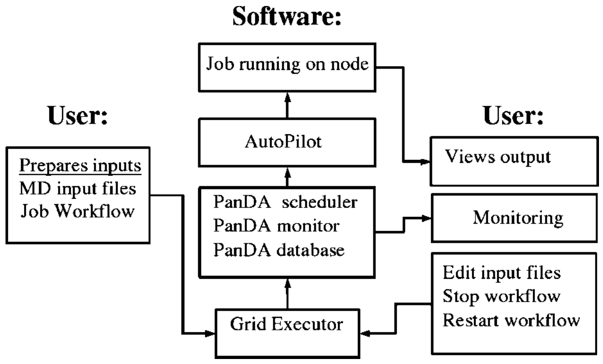


Figure 1. Basic steps in running MD simulations on the OSG.

The utility of this approach to grid computing is demonstrated by studying the hydration of the interior of a protein. Previously, we have shown that even a seemingly trivial question, determination of populations of two rotameric states of a Glu side chain located in the protein interior and hydration of this side chain, poses a nontrivial sampling problem.[24] Here, we employ grid computing to provide enhanced sampling and study the coupling between conformations and hydration of an internal Glu side chain. In the framework of this study, grid computing is also used to benchmark a relatively new method for enhanced conformational search, the self-guided Langevin dynamics,[6] against the more traditional Langevin dynamics method.

METHODS

The steps involved in running MD jobs via the OSG resource are represented in Figure 1. The user prepares the input files for MD runs and designs the job workflow, which is passed to a custom software program called the grid executor. To be allowed to run jobs on OSG resources, the user has to become a member of an OSG Virtual Organization (VO). The user must properly authenticate with their certificate to allow the grid executor to submit jobs on their behalf. Once it has parsed the input file, the grid executor submits and tracks the jobs required to complete the workflow. When executing on the grid, a transformation script downloads the executable and input file, runs CHARMM, copies results off, and cleans up all files on the nodes. The output files are copied back to the user's home computer, although the software allows for a grid storage resource to be used instead, provided that one has been properly configured. For this implementation, a middleware known as PanDA[25] was used to provide job scheduling and monitoring functions. The user can monitor jobs either by looking at the output files on his or her computer or by visiting the PanDA webpage which displays the status of all jobs. All of these steps are discussed in more detail below.

User Prepared Files and Workflow. In a first step, the user prepares files necessary for molecular dynamics simulations. In the application represented in this paper, the CHARMM program[23] was used, but in principle, any program for molecular dynamics runs, such as AMBER,[26] NAMD,[27] GROMACS,[28] etc., can be used with the minor modification to the existing software. The necessary files usually contain the MD program executable, the coordinate and structure files of the simulated system, the force-field topology and parameter files, and the input files that specify the stages and details of the MD runs.

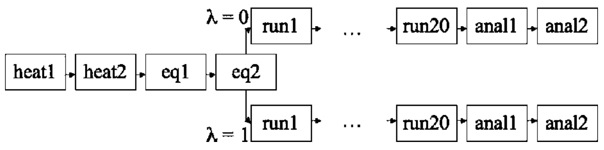


Figure 2. Description of the workflow used in the calculations.

A standard MD job consists of several stages. Heating of the system to a desired temperature is the first step. This stage will be referred to as "heat". Equilibration, referred to as "eq", of the system at the desired temperature follows. Next come the production runs, referred to as "run". Several types of analysis jobs may follow.

The run stage takes the longest to complete. For example, 1 ns of MD runs on a single processor may take about 1 month to complete. Many sites on the OSG limit jobs to a certain amount of wall time. Thus, the long dynamics calculation must be split into a large number of small jobs, each taking no more than 12 h. Often the heating and equilibration steps need to be split into several jobs as well. In the application used here, we had two heating jobs, two equilibration jobs, and 40 run jobs. Furthermore, in order to test two methods to run molecular dynamics, each equilibration job branched into two sets of run jobs, where each run job was performed with different parameters. Finally, for each run job, two analysis jobs were performed. The workflow describing all simulation stages is shown in Figure 2.

To specify the details of the workflow, the user has to create a file readable by the grid executor, describing the different jobs that have to be run to complete the task. A sample input file in which the workflow is specified are given in the Supporting Information. The user must also edit a shell script which defines where various input and output files are stored. The grid executor is described in more detail below.

Open Science Grid. The OSG is a loosely coupled system that spans multiple organizations, and thus, advanced software is necessary to manage resources, dispatch work, and ensure security. The OSG provides a common set of tools, to ensure a standardized interface to users' jobs. This common set of tools, known as the Virtual Data Toolkit (VDT), provides grid services such as the Condor batch system as well as other utilities, especially those that enable management and access to dispersed data. The VDT also includes the Globus toolkit, one of the most popular grid middleware products.

Grid users are members of one or more virtual organizations (VOs). A VO is a group of users working on a related project or group of projects that require support for specific software and data. Sites make decisions to support a given VO based on the hardware and software that they are willing to provide and maintain. VOs also play an important role in maintaining the security of the grid by vetting persons who want permission to use grid resources and approving requests for grid certificates. Authentication and access control services are provided via a public key infrastructure conformant to the X509 standard. For this work, calculations were performed as part of the Open Science Grid VO, which exists to help transition new users and applications onto running on the grid. Information about how users can join the Open Science Grid can be found in the Supporting Information.

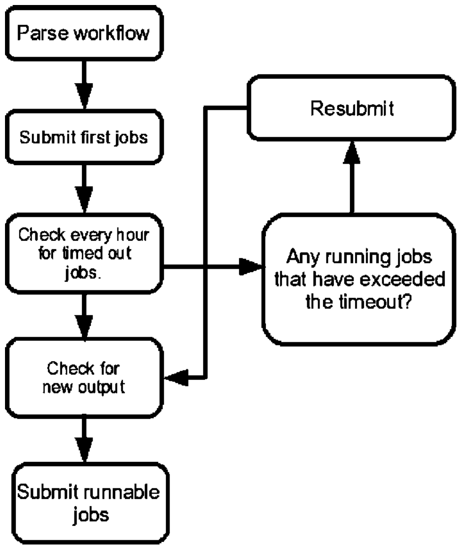

Parse workflow
Submit first jobs
Check every hour for timed out jobs.
Check for new output
Submit runnable jobs
Resubmit
Any running jobs that have exceeded the timeout?

Figure 3. Architecture of the grid executor.

PanDA. Jobs are usually submitted to the grid and monitored via a grid enabled version of Condor.[29] In order to simplify this process, the PanDA[25] workload management and monitoring system was developed. PanDA uses software called AutoPilot, which submits "pilot jobs" to the grid that run basic checks on a node to ensure that the software configuration required by the job is in place and, then, request a workload job to run. In theory, such checks are done by the site administrator for each VO supported by the site. However, given the complex nature of scientific software, there is ample opportunity for seemingly unrelated software changes or operating system or library problems to interfere with the successful completion of a job. AutoPilot runs a simple script on the node to determine as much as possible about the configuration and available consumable resources (e.g., memory and local hard disk scratch space) on the node and thereby determine if it is capable of running the user's job. The exact nature of the script run and tests performed is dependent on the exact job type. In this way, jobs are not sent to the grid for processing until a validated processor capable of running them has been acquired. A second advantage to using PanDA is that it provides a Web based monitor that shows job progress and errors.

Grid Executor. For this project, there was a need to simultaneously submit and monitor a large number of jobs. To accomplish this, a wrapper around PanDA called the grid executor was developed. It has the ability to read in a description of a workflow and then start, monitor, and restart the jobs required to complete that workflow. This saves the end user from having to start and monitor each job individually, while still allowing them to use PanDA's monitoring software to get more details about each job run. In addition to the grid executor, there is a small CHARMM wrapper script which is run on the grid execution node that fetches the CHARMM executable and the inputs, runs CHARMM, and copies the output back.

The grid executor is a daemon written in Python, which runs on the user's local workstation or on a computer to which the user has a high bandwidth network connection. The architecture of the executor is shown in Figure 3. The executor parses the file in which the workflow is described and makes a list of all the jobs that need to be run, what input files they need, and what output files they produce. Internally, individual jobs are represented as Python objects. Each job object is created with a list of jobs on which it dependents, the files that it requires as input and produces as output, and state information.

When all individual job objects are created, the executor launches the first set of jobs. It then goes into a loop to wait for output files to be produced. The jobs executing on the grid nodes copy their output to a specific place accessible to the executor. When the executor discovers a new file in that directory, it marks the associated job as complete and checks to see if any new jobs are runnable. If so, they are launched. The executor keeps track of which jobs have been launched and how long they have been running for. If a job does not report results back within a set time (currently 60 h), it is assumed to have failed and is resubmitted. If a job fails ten times, it is assumed to be user error and further efforts to run it are abandoned.

The 60 h timeout was chosen because the jobs for this project were benchmarked as taking approximately 12 h to run on the nodes at most sites. The remaining 48 h were chosen to allow enough time for jobs to sit in queues and prevent swamping of the grid with resubmissions of the same job. This scheme worked well but could be improved by having the executor itself query the PanDA monitor on the status of jobs so that it could restart failed ones automatically, rather than waiting for the timeout to expire. The end user may force a job to restart by stopping the executor, manually editing its state file, and restarting it (see below).

Once every hour, the grid executor writes out a state file. This file has three parts: a header, a list of completed jobs, and a list of running jobs with the times that they were started. This allows the executor to be stopped and restarted. The state file is just an ASCII text file, allowing an advanced user to force jobs to rerun or modify the workflow on the fly. A sample state file is given in the Supporting Information. The executor also produces a log file which details the exact actions it is performing.

User Monitoring. The status of all jobs on the grid is available through the PanDA website. This site lists the jobs which are running, which have finished successfully, and which have failed. A screenshot from the PanDA Monitor is shown in Figure 4. The output of jobs that have terminated (successfully or unsuccessfully) as well as the debugging information is also accessible through this website. The CHARMM wrapper script prints out debugging information which is accessible through the PanDA website. This script ensures that CHARMM terminates successfully before copying the results out to the designated machine. The users may then monitor completed jobs by looking at the log files of successfully terminated jobs.

Users may modify the workflow by editing the state file and restarting the grid executor. Because the input files are downloaded separately for each job, it is more simple to modify them and force selected jobs to rerun. However, care is needed when making radical changes to an executing workflow as the executor expects the total number of threads to remain the same. Integrating workflow rearrangement and monitoring on the part of the grid executor more tightly into PanDA would be a desirable feature to implement in the future.

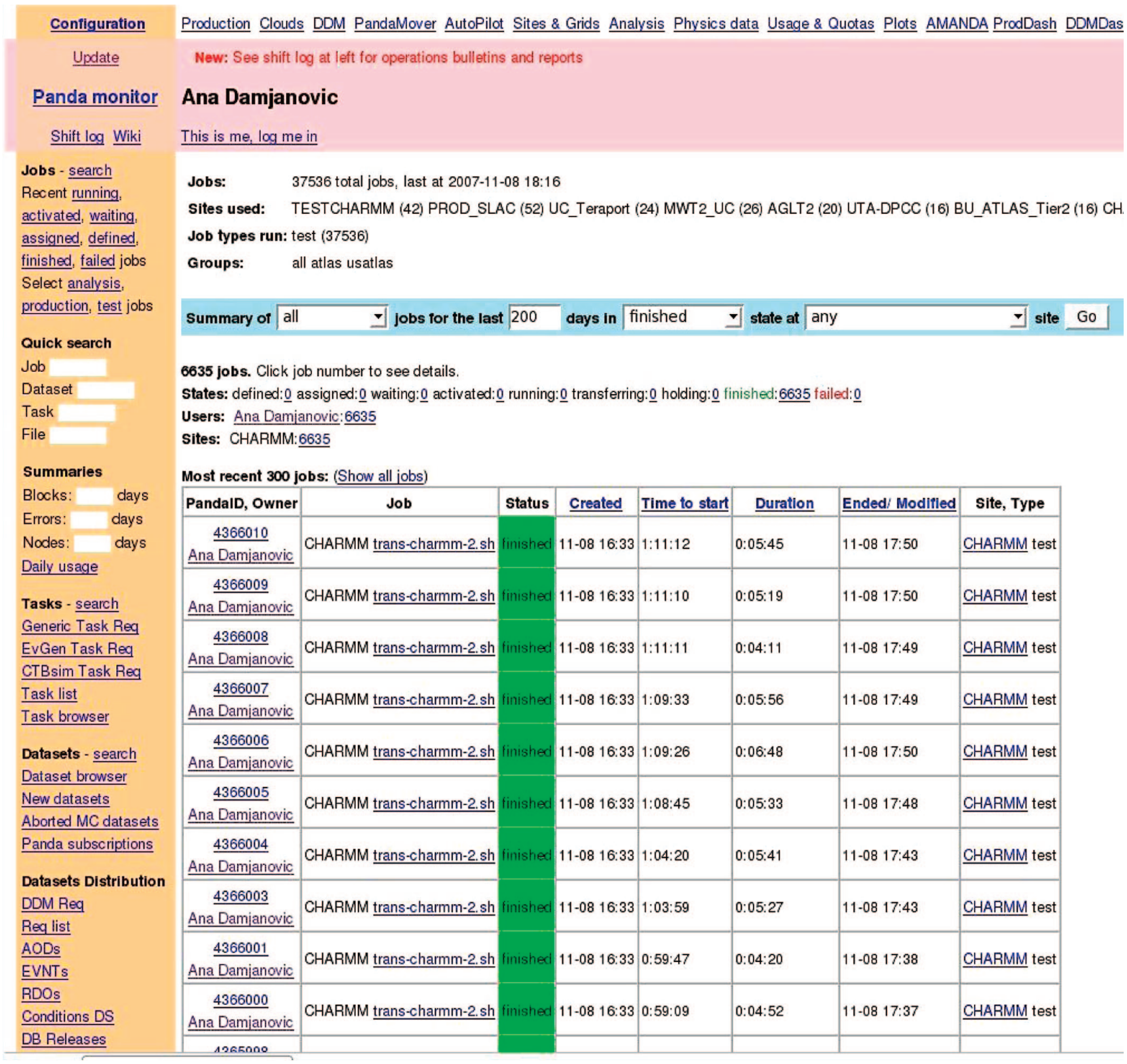
INITIAL APPLICATION: HYDRATION OF THE PROTEIN INTERIOR

Water molecules in the interior of proteins play important functional roles in biochemical precesses such as catalysis or proton transfer. They can influence the protein's physical properties, such as stability, flexibility, or effective dielectric constant. Despite their biological importance, the exact number and locations of water molecules in the interior of proteins are not always known. This uncertainty stems from the fact that disordered water molecules are usually invisible in crystallographic experiments. Significant insights about structural, dynamical, and thermodynamic properties of interior and surface water molecules can be gained from computational studies.[30−34]

Variants of staphylococcal nuclease (SNase) with internal hydrophobic residues substituted with polar or ionizable ones[35−39] were used to study disordered water molecules in the protein interior. The SNase variant with internal Glu-66 was particularly interesting for this purpose. The pK_a of Glu-66 is 8.8,[36] and this Glu residue is uncharged at physiological pH. The crystallographic structure of SNase with Glu-66 obtained at cryogenic temperatures revealed two water molecule buried in association with the carboxylic oxygen atoms of Glu-66 (Figure 5). The structure obtained at room temperature showed only one internal water molecule. Molecular dynamics simulations of this variant showed that the number of water molecules associated with Glu-66 is dependent on the conformation of the Glu-66 side chain.[24,40] When the side chain is in an extended (or straight) conformation, one or two water molecules are associated with it. When the side chain is twisted and turned toward the protein interior, the carboxylic group of Glu-66 interacts with the polar atoms of the backbone in the protein core, the carboxylic group is dehydrated. Snapshots from MD simulations representative of the straight and the twisted conformation of the Glu-66 side chain are shown in Figure 5.

The crystallographic structures showed no density in the region of space corresponding to the twisted conformation of the Glu-66 side chain, suggesting that the population of this state in the crystal is small. The factors that may influence the degree of agreement between simulations and experiments have recently been discussed in ref 41. Some of the factors include quality of force-fields[42] and sampling. The sampling provided by the nine 10 ns long MD simulations and one 50 ns long simulation performed previously[24] was insufficient to estimate the population of the twisted state of the Glu-66 side chain. As the results appeared to be sensitive to assigned initial velocities, in this paper, we opted to estimate the population of this state by running many short simulations. To investigate how sensitive the results were to the number of internal water molecules included in the starting structure, simulations started with different initial hydration states (IHSs) were performed and compared. In addition, using another set of simulations, the performance and accuracy of a new method for conformational search, the self-guided Langevin dynamics method (SGLD) was tested through comparison with simulations obtained with the regular Langevin dynamics.

Simulated Systems, the Workflow, and the SGLD Method. The crystal structure of the PHS/V66E[36] variant of staphylococcal nuclease was used as a starting structure



New: See shift log at left for operations bulletins and reports

Ana Damjanovic

This is me, log me in

Jobs: 37536 total jobs, last at 2007-11-08 18:16
Sites used: TESTCHARMM (42) PROD_SLAC (52) UC_Teraport (24) MWT2_UC (26) AGLT2 (20) UTA-DPCC (16) BU_ATLAS_Tier2 (16) CH
Job types run: test (37536)
Groups: all atlas usatlas

Summary of all jobs for the last 200 days in finished state at any site Go

6635 jobs. Click job number to see details.
States: defined:0 assigned:0 waiting:0 activated:0 running:0 transferring:0 holding:0 finished:6635 failed:0
Users: Ana Damjanovic:6635
Sites: CHARMM:6635

Most recent 300 jobs: (Show all jobs)

PandaID, Owner	Job	Status	Created	Time to start	Duration	Ended/ Modified	Site, Type
4366010 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:11:12	0:05:45	11-08 17:50	CHARMM test
4366009 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:11:10	0:05:19	11-08 17:50	CHARMM test
4366008 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:11:11	0:04:11	11-08 17:49	CHARMM test
4366007 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:09:33	0:05:56	11-08 17:49	CHARMM test
4366006 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:09:26	0:06:48	11-08 17:50	CHARMM test
4366005 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:08:45	0:05:33	11-08 17:48	CHARMM test
4366004 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:04:20	0:05:41	11-08 17:43	CHARMM test
4366003 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	1:03:59	0:05:27	11-08 17:43	CHARMM test
4366001 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	0:59:47	0:04:20	11-08 17:38	CHARMM test
4366000 Ana Damjanovic	CHARMM trans-charmm-2.sh	finished	11-08 16:33	0:59:09	0:04:52	11-08 17:37	CHARMM test

Figure 4. Monitoring of jobs via the PanDa webpage.

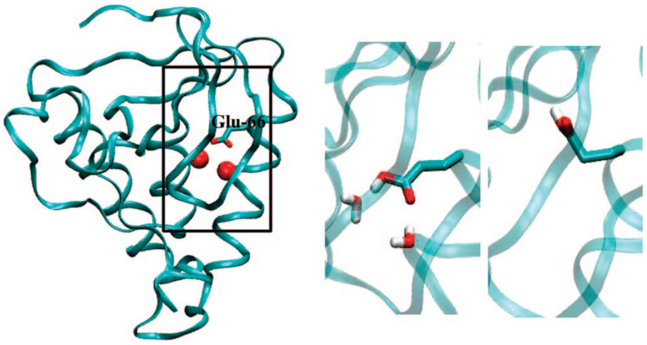


Figure 5. (left) Crystal structure of the V66E variant. (right) Snapshots from MD simulations representative of the straight (left) and the twisted (right) conformation of the Glu-66 side chain.

for the simulations. The IHS S0 was created by deleting two water molecules that hydrate the Glu-66 side chain from the initial structure. In contrast, the IHS S2 contained the two crystallographic water molecules.

For the system setup and subsequent MD and SGLD runs, the program CHARMM[23] was used, in conjunction with the CHARMM27 force field.[43] In addition to the protein, the final systems contained 8 Na^+ ions, 17 Cl^- ions, 5158 water molecules in the S0 structure, and 5171 water molecules in the S2 structure. The details of the system setup and Langevin dynamics runs are the same as described elsewhere.[2] For each of the simulated systems (S0 and S2), 40 heating and equilibration runs were performed, with different seed numbers for the random number generator used for assigning initial velocities. For each of the equilibrated systems, a Langevin dynamics (LD) and a self-guided Langevin dynamics (SGLD) run was performed.

The equation of motion for the SGLD dynamics of an N particle system is

$$\dot{\vec{p}}_i = \vec{f}_i - \gamma_i \vec{p}_i + \vec{R}_i + \lambda_i \gamma_i \langle \vec{p}_i \rangle_L \qquad (1)$$

The first three terms in this equation define the typical Langevin dynamics, where γ_i is the collision frequency, and $\vec{R}_i$ is a random force. The last term in the equation represents the so-called guiding force which is used to enhance systematic conformational changes. The parameter λ_i is the guiding factor which controls the strength of the enhancement. The local averaging time, t_L, defines the slow motions that are to be enhanced. It is determined as $t_L = L\delta t$, where δt is the time step and L is the number of timesteps over which the averaging is performed.

The same friction coefficient γ and guiding factor λ was used for all particles. The SGLD parameters $t_L = 0.1$ ps, and $\lambda = 1.0$. were used. The friction coefficient was $\gamma = 1$ 1/ps for both Langevin dynamics ($\lambda = 0.0$) and SGLD. The weak friction is used here to control temperature and enhance conformational transitions rather than to mimic the effect of the solvent which is explicitly included in the simulations. Because of the guiding component of these simulations, information about the timescales of protein dynamics is lost.

Protein and Water Conformations. Statistical analysis of the populations of the rotameric states of Glu-66 and hydration of the side chain is shown in Table 1. Two major rotameric states of Glu-66 were observed: "twisted", characterized by a CA−CB−CG−CD dihedral angle of about 70°, and "straight", characterized by a CA−CB−CG−CD dihedral angle of about 180°.

Comparison of populations of the two rotameric states during the first and the second nanoseconds of the simulation (Table 1) suggests that the systems are still in the process of reaching equilibrium. The populations of the two conformations are not fully converged, and the results presented yield a qualitative rather than quantitative picture. In all simulations, the straight conformation became favored with time. This agrees with experimental electron density which is low in the area of the twisted conformation. With time, the hydration of the side chain increases or stays the same in all simulations.

Table 1 confirms that the straight conformation is more hydrated than the twisted conformation. Large hydration of the side chain in the straight conformation is consistent with the crystal structure which shows one water molecule (room T structure)[38] or two water molecules (cryogenic structure)[36] around the straight Glu-66 side chain. Table 1 suggests that during the second nanosecond, water molecules can penetrate the protein interior even when Glu-66 is in a twisted state, i.e, when the polar group of Glu-66 is far away from bulk water. A previous study[40] has shown that one (or briefly more than one) water molecule can penetrate the interior even in the absence of polar side chains in the interior. Such water molecules can interact with polar groups of the protein backbone or with other water molecules.

Comparison of Simulations with Different Initial Hydration States. Protein and water populations derived from simulations started with initial hydration states S0 and S2 differ substantially, as shown in Table 1. S2 favors the straight conformation, on average by 25% of the time during the first ns and by 20% of the time during the second ns. As the difference in results of the simulations with the two IHSs becomes smaller with time, the results seem to be converging in the same direction.

Table 1. Percentage of Total Simulation Time that Glu-66 was in the Straight or Twisted Conformation[a]

	Glu-66 straight					Glu-66 twisted				
		# of water molecules					# of water molecules			
system	% time	0	1	2	3	% time	0	1	2	3
					first nanosecond					
S0, $\lambda = 0$	57	63	30	7	0	43	100	0	0	0
S0, $\lambda = 1$	49	71	22	7	0	51	100	0	0	0
S2, $\lambda = 0$	78	34	47	18	1	22	100	0	0	0
S2, $\lambda = 1$	76	37	43	20	0	24	100	0	0	0
					second nanosecond					
S0, $\lambda = 0$	68	47	39	14	0	32	94	6	0	0
S0, $\lambda = 1$	55	68	20	12	0	45	96	4	0	0
S2, $\lambda = 0$	84	27	54	18	0	16	90	8	2	0
S2, $\lambda = 1$	79	28	40	30	1	21	100	0	0	0

[a] Percentage of total simulation time that the number of water molecules found with a 3.5 Å radius of Glu-66 carboxylic oxygens was 0, 1, 2, or 3. Water molecules were counted separately for the conformations when Glu-66 side chain was straight and twisted.

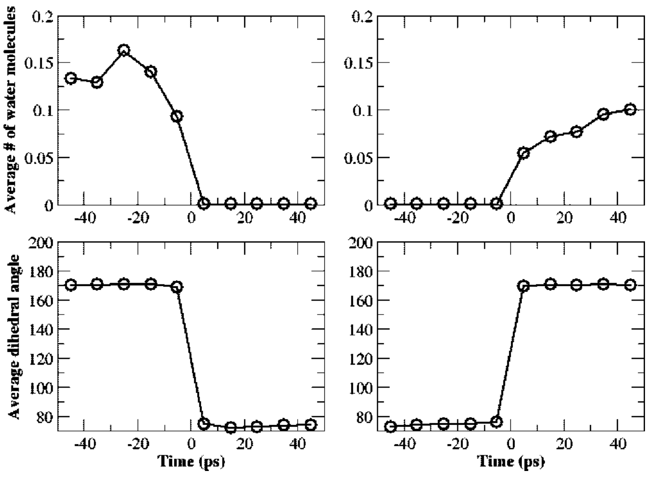


Figure 6. (top) Average number of water molecules within 3.5 Å of Glu-66 carboxylic oxygens, 50 ps before and after conformational transition of Glu-66. (bottom) Average CA−CB−CG−CD dihedral angle of Glu-66. The dihedral angle was wrapped to the 0−π interval. (left) Transitions from straight into twisted state. (right) Transitions from twisted to straight state.

Comparison of the number of water molecules around the Glu-66 side chain in Table 1 suggests that the straight side chain is dehydrated significantly more in the simulations with the S0 IHS. The dehydrated side chain in S0 simulations is preferred by the same amount (about 30% of the time) during the first and second nanoseconds of simulations.

Correlation between Average Number of Bound Water Molecules and Dihedral Angle of the Side Chain. To estimate whether dehydration is responsible for twisting of the Glu-66 side chain, we have determined the average number of water molecules around the Glu-66 side chain, 50 ps before and 50 ps after the side chain switched conformations. The averaging was performed over all recorded transitions provided that no other transitions occurred within the 50 ps time period before and after the transition. The averaging was performed for a sample of 40 2 ns long simulations for which we recorded timesteps every picosecond. The values shown in Figure 6 are averages over 10 ps time periods.

Figure 6 shows two interesting results. One is that the average number of water molecules during 50 ps before the transition from straight into twisted conformation is 0.13.

This is much smaller that the average number of water molecules around the Glu-66 side chain when it is straight, which, for the studied sample of 40 2 ns long simulations, is 0.8. Inspection of the nine trajectories in which the conformation of Glu-66 remained straight during the entire 2 ns run revealed that the average number of water molecules is 1.37 and that the number of water molecules around Glu-66 is 3 for 2% of the time, 2 for 37% of the time, 1 for 58% of the time, and 0 for 3% of the time. On the other hand, the number of water molecules during 50 ps before the transition of Glu-66 from straight into twisted conformation fluctuates between 0 and 1. In 29 out of the 45 registered occurrences of such transitions, the number of water molecules remained 0 during the entire 50 ps time period before the switch.

Figure 6 further suggests that there is a decrease in the number of water molecules immediately before the transition from the straight into the twisted conformation, as the average number of water molecules during the last 10 ps before the transition is smaller that the average number of water molecules during 40 ps before that. This is an indication that there is a temporal correlation between dehydration of the side chain and conformational transition. To estimate an approximate rate at which dehydration of the side chain induces conformational transition, we counted the number of transitions from 1 to 0 water molecules around the Glu-66 side chain. Only events in which the side chain remained dehydrated for longer than 30 ps were considered. A total of 109 such transitions were counted based on the 40 2 ns long trajectories. Out of those 109, 12 resulted in a switch of the Glu-66 dihedral angle.

In the case of transitions from the twisted into the straight conformation of Glu-66, Figure 6 indicates that the number of water molecules hydrating the side chain of Glu-66 increases gradually, suggesting that rehydration is perhaps a Poisson process after a transition.

Comparison of LD and SGLD. Simulations started with LD and SGLD are compared in Table 1. The twisted state is favored by SGLD, but only by a small difference (on average, by 7%). In terms of the hydration, the dehydrated state is favored, again by only a small percent (8%). Previously it has been shown that the frequency of observation of α-helix and *β*-strand fraying is about twice as large in SGLD than in LD.[2] The number of hops between the two conformations of the side chain of Glu-66, based on all simulations, is only slightly larger in SGLD than in LD (228 vs 191). This is somewhat surprising as SGLD has been shown to increase the frequency of barrier crossing.[6] Since conformational transitions are influenced by dehydration, perhaps in this case dehydration is not limited by an enthalpic barrier. SGLD is not supposed to increase the frequency of such processes.

Conclusions. Simulations using the Open Science Grid were used to study hydration of the side chain of the internal Glu-66 in staphylococcal nuclease. Previous studies[24] have revealed that Glu-66 can exist in two conformations, straight and twisted, but the exact populations of the conformations could not be calculated in a statistically significant manner. Rather than extending these simulations further, in this paper we approach this problem by running 40 2 ns long simulations by taking advantage of the resources of the OSG. In addition, we studied the dependence of the results on the initial hydration state (IHS) and evaluated the performance of the relatively new SGLD method for enhanced conformational sampling. In total, 4 sets (2 IHS that branch into LD and SGLD runs) of 40 simulations were performed and analyzed.

The 40 2 ns long simulations were not enough for the results to converge. This precludes a quantitative analysis of the hydration of the Glu-66 side chain but allowed the observation of the coupling between protein conformations and the state of hydration.

The initial state of hydration (IHS) seems to have a large impact on the relative populations of the two side-chain conformations. The simulations with the IHS that did not include the two water molecules observed in the crystal structures yielded populations of the twisted side chain that were 20% larger than the simulations with the IHS that included the two crystallographic water molecules. Because no electron density corresponding to the twisted conformation of Glu-66 was observed in a crystallographic studies,[38] the simulations with the IHS that contained the crystallographic water were more accurate. More importantly, the disagreement in the results from simulations started with the two IHS suggests that the water content can bias protein conformations.

Analysis of the correlations between water content and transitions from the straight into the twisted conformation suggests that water content can influence these transitions. When the side chain is well hydrated, i.e, when there is on average more than 1 water molecule around the side chain, transitions into the twisted state did not occur. These transitions occur when there is, on average, 0.13 water molecules around the side chain. Furthermore, we reveal that there is a decrease in the number of water molecules around the side chain shortly (on the order of picoseconds) before the conformational transition, suggesting that dehydration of the side chain is one of the driving forces for the conformational transition.

The results of simulations with LD and SGLD methods yield side chain and water populations that agree up to 8%. The SGLD yields a slightly larger number of hops between the two conformations. Even though it seems that the simulations with the dehydrated IHS, and with SGLD, are converging in the same direction as the simulations with hydrated IHS and LD, further grid calculations are needed to test whether the same final populations will be obtained.

OUTLOOK

The broader goal of this project is to promote grid computing among the general scientific community. To gain access to the OSG resources, a science researchers will need to join a virtual organization (VO). A VO dedicated to biomolecular simulations will soon be formed and will be named BioMolSim VO. The grid executor software, developed for the purpose of user-friendly submission, execution, and monitoring of grid jobs, will be made available to the members of the BioMolSim VO. Eventhough the grid executor is developed to work with the molecular dynamics and modeling program CHARMM, it can easily be modified to work with any other molecular dynamics software, e.g., AMBER, NAMD, GROMACS, etc.

Improvements are planned for the future versions of the grid executor. Presently, CHARMM runs on single processor machines. MPI enabled CHARMM is currently being tested on the OSG. Because of the importance of speeding up the simulations through parallel jobs, it is our priority to make the parallel version of CHARMM operational as soon as possible.

Support of more robust workflows with complex syntax/decision making capabilities is planned. This will allow on-the-fly analysis of the simulations for a certain desired criterion (e.g., a conformational transition) and automated enhancement of the simulations that meet the criterion.

We are also planing to better integrate the executor with PanDA and the rest of the OSG software stack (e.g., the Condor queues at sites). This will shorten the times that jobs are sitting in queues and the overall simulation time. Finally, we will make it easier for the scientist to modify how and where files are stored.

Although challenges remain, the grid software stake has proved amenable to large scale molecular simulation. Computational chemists are beginning finally to take advantages of resources previously available primarily for the physics community.

ACKNOWLEDGMENT

This research was done using resources provided by the Open Science Grid, which is supported by the National Science Foundation and the U.S. Department of Energy's Office of Science. We thank all of the institutions that have provided computers for this project. We would like to thank Ruth Pordes, Frank Würthwein, and Alain Roy for their assistance with the Open Science Grid, and Rich Pastor for useful discussions. This work was also supported by NIH grant (R01 GM061597) to B.G.-M.E., the Intramural Research program at NIH, NHLBI, and the A. P. Sloan foundation (BR-4379) and NSF grant (PHY-0457374) to P.M.

Supporting Information Available: More information on how to join the Open Science Grid, as well as sample files necessary for running of CHARMM jobs on the OSG. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Duan, Y.; Kollman, P. A. Computational protein folding: from lattice to all-atom. *Science* **2001**, *40*, 297–309.

(2) Damjanović, A.; Wu, X.; García-Moreno E., B.; Brooks, B. R. Backbone relaxation coupled to the ionization of internal groups in proteins: A self-guided Langevin dynamics study. *Biophys. J.*, published online July 18, 2008, http://dx.doi.org/10.1529/biophysj.108.130906.

(3) Elber, R. Long-timescale simulation methods. *Curr. Op. Struct. Biol.* **2005**, *15*, 151–156.

(4) Sugita, Y.; Okamoto, Y. Replica-exchange multicanonical algorithm and multicanonical replica-exchange method for simulating systems with rough energy landscape. *Chem. Phys. Lett.* **2000**, *329*, 261–270.

(5) Sanbonmatsu, K. Y.; García, A. E. Structure of met-enkephalin in explicit aqueous solution using replica exchange molecular dynamics. *Proteins: Struct., Func., Gen.* **2002**, *46*, 225–234.

(6) Wu, X. W.; Brooks, B. R. Self-guided Langevin dynamics simulation method. *Chem. Phys. Lett.* **2003**, *381* (3−4), 512–518.

(7) Shirts, M.; Pande, V. J. Screen savers of the world unite. *Science* **2000**, *290* (5498), 1903–1904.

(8) Taufer, M.; An, C.; Kerstens, A.; Brooks, C., III. Predictor@home: A protein structure prediction supercomputer based on global computing. *IEEE Trans. Parallel Distrib. Syst.* **2005**, 200b.

(9) Bradley, P.; Malmstrom, L.; Qian, B.; Schonbrun, J.; Chivian, D.; Kim, D. E.; Meiler, J.; Misura, K. M.; Baker, D. Free modeling with rosetta in casp6. *Proteins: Struct., Func., Gen.* **2005**, *61*, 128–134.

(10) Foster, I. The grid: A new infrastructure for 21st century science. *Phys. Today* **2002**, *55* (2), 42–47.

(11) Pordes, R.; Petravick, D.; Kramer, B.; Olson, D.; Livny, M.; Roy, A.; Avery, P.; Blackburn, K.; Wenaus, T.; Würthwein, F.; Foster, I.; Gardner, R.; Wilde, M.; Blatecky, A.; McGee, J.; Quick, R.; on behalf of the Open Science Grid consortium. The open science grid. *J. Phys.: Conf. Ser.* **2008**, *78*, 012057.

(12) Catlett, C.; Allcock, W. E.; Andrews, P.; Aydt, R.; Bair, R.; Balac, N.; Banister, B.; Barker, T.; Bartelt, M.; Beckman, P.; Berman, F.; Bertoline, G.; Blatecky, A.; Boisseau, J.; Bottum, J.; Brunett, J.; Bunn, J.; Butler, M.; Carver, D.; Cobb, J.; Cockerill, T.; Couvares, P. F.; Dahan, M.; Diehl, D.; Dunning, T.; Foster, I.; Gaither, K.; Gannon, D.; Goasguen, S.; Grobe, M.; Hart, D.; Heinzel, D.; Hempel, C.; Huntoon, W.; Insley, J.; Jordan, C.; Judson, I.; Kamrath, A.; Karonis, N.; Kesselman, C.; Kovatch, P.; Lane, L.; Lathrop, S.; Levine, M.; Lifka, D.; Liming, L.; Livny, M.; Loft, R.; Marcusiu, D.; Marsteller, J.; Martin, S.; McCaulay, S.; McGee, J.; McGinnis, L.; McRobbie, M.; Messina, P.; Moore, R.; Moore, R.; Navarro, J. P.; Nichols, J.; Papka, M. E.; Pennington, R.; Pike, G.; Pool, J.; Reddy, R.; Reed, D.; Rimovsky, T.; Roberts, E.; Roskies, R.; Sanielevici, S.; Scott, J. R.; Shankar, A.; Sheddon, M.; Showerman, M.; Simmel, D.; Singer, A.; Skow, D.; Smallen, S.; Smith, W.; Song, C.; Stevens, R.; Stewart, C.; Stock, R. B.; Stone, N.; Towns, J.; Urban, T.; Vildibill, M.; Walker, E.; Welch, V.; Wilkins-Diehr, N.; Williams, R.; Winkler, L.; Zhao, L.; Zimmerman A. Teragrid: Analysis of organization, system architecture, and middleware enabling new types of applications. In *HPC and Grids in Action*; Grandinetti, L., Ed.; Advances in Parallel Computing; IOS Press: Amsterdam, 2007.

(13) Lesyng, B.; Bala, P.; Erwin, D. Eurogrid: European computational grid testbed. *J. Parallel Distrib. Comput.* **2003**, *63*, 590–596.

(14) Pytlinski, J.; Skorwider, L.; Benedyczak, K.; Wronski, M.; Bala, P.; Huber, V. Uniform access to the distributed resources for the computational chemistry using unicore. In *Computational Science−ICCS2003*; Lecture Notes in Computer Science; Springer-Verlag: London, U.K.; 2003; p 686.

(15) Choi, Y.; Kim, S.-R.; Hwang, S.; Jeong, K. A grid computing-based monte carlo docking simulations approach for computational chiral discrimination. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*; Lecture Notes in Computer Science; Springer: Berlin, Germany; 2005; pp 446−455.

(16) Jacq, N.; Breton, V.; Chen, H.-Y.; Ho, L.-Y.; Hofmann, M.; Lee, H.-C.; Legré, Y.; Lin, S.-C.; Maass, A.; Medernach, E.; Merelli, I.; Milanesi, L.; Rastelli, G.; Reichstadt, M.; Salzemann, J.; Schwichtenberg, H.; Sridhar, M.; Kasam, V.; Wu, Y.-T.; Zimmermann, M. Large scale in silico screening on grid infrastructures. In *Proceedings of The Third International Life Science Grid Workshop, LSGrid 2006*, Yokohama, Japan, October 13–14, 2006.

(17) Sild, S.; Maran, U.; Lomaka, A.; Karelson, M. Open computing grid for molecular science and engineering. *J. Chem. Inf. Model.* **2006**, *46*, 953–959.

(18) Natrajan A.; Crowley M.; Wilkins-Diehr, N.; Humphrey M.; Fox A.; Grimshaw A.; Brooks,C., III. Studying protein folding on the grid: Experiences using charmm on npaci resources under legion. In *Proceeding of the HPDC Conference*, San Francisco, CA, August 7–9, 2001.

(19) Gower, M.; Cohen, J.; Philips, J.; Kufrin, R.; Schulten, K. Managing biomolecular simulations in a grid environment with namd-g. In *Proceedings of the 2006 TeraGrid Conference*, Indianapolis, IN, June 13–15, 2006.

(20) Woods, C. J.; Ng, M. H.; Johnston, S.; Murdock, S. E.; B.; Wu, K. T.; Fangohr, H.; Jeffreys, P.; Cox, S.; Frey, J. G.; Sansom, M. S.; Essex, J. W. Grid computing and biomolecular simulation. *Phil. Trans. R. Soc. Lond. A* **2005**, *262*, 2017–2035.

(21) Wang, W.; Chen G.-L.; Chen, H.-P.; Yang, S. A grid computing framework for large scale molecular dynamics simulations. In *Grid and Cooperative Computing*; Lecture Notes in Computer Science; Springer: Berlin, Germany; 2004; pp 645−648.

(22) Wilter, A.; Osthoff, C.; Oliveira, C.; E. B. Gomes, D.; Hill, E.; Dardenne, L. E.; Barros, P. M.; A. A. G. L. Loureiro, P.; Novaes, R.; Pascutt, P. G. The biopauá project: A portal for molecular dynamics using grid environment. In *Advances in Bioinformatics and Computational Biology*; Lecture Notes in Computer Science; Springer: Berlin, Germany; 2005; pp 214−217.

(23) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **1983**, *4*, 187–217.

(24) Damjanović, A.; Schlessman, J. L.; Fitch, C. A.; García, A. E.; García-Moreno E., B. Role of flexibility and polarity as determinants of the hydration of internal cavities and pockets in proteins. *Biophys. J.* **2007**, *93* (8), 2791–2804.

(25) Maeno, T. Panda: Distributed production and distributed analysis system for atlas. In *Proceedings of Computing in High Energy Physics 2007*, September 2–7, 2007.

(26) Case, D.; Cheatham, T. E.; Darden, T.; H.; Gohlke.; Luo, R.; Merz, K., Jr.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. The amber biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(27) Kalé, L.; Skeel, R.; Bhandarkar, M.; Brunner, R.; Gursoy, A.; Krawetz, N.; Phillips, J.; Shinozaki, A.; Varadarajan, K.; Schulten, K. NAMD2: Greater scalability for parallel molecular dynamics. *J. Comp. Phys.* **1999**, *151*, 283–312.

(28) Lindhal, E.; Hess, B.; van der Spoel, D. Gromacs 3.0: A package for molecular simulation and trajectory analysis. *J. Mol. Mod.* **2001**, *7* (8), 306–317.

(29) D.Thain, T. T.; Livny, M. Condor and the grid. In *Grid Computing: Making The Global Infrastructure a Reality*; Berman, F., J. G. Hey, A., Fox, G. Eds.; John Wiley: New York, 2003.

(30) Wade, R. C.; Mazor, M.; McCammon, J. A.; Quiocho, F. A. A molecular dynamics study of thermodynamic and structural aspects of the hydration of cavities in proteins. *Biopolymers* **1991**, *31*, 919–31.

(31) Steinbach, P. J.; Brooks, B. R. Protein hydration elucidated by molecular dynamics simulation. *Proc. Natl. Acad. Sci. USA* **1993**, *90*, 9135–9139.

(32) Makarov, V. A.; Andrews, B. K.; Smith, P. E.; Pettit, B. M. Residence times of water molecules in the hydration sites of myoglobin. *Biophys. J.* **2000**, *79*, 2966–2974.

(33) García, A. E.; Hummer, G. Water penetration and escape in proteins. *Proteins: Struct., Func., Gen.* **2000**, *38*, 261–272.

(34) Woolf, T. B.; Grossfield, A.; Tychko, M. Differences between apo and three holo forms of the intestinal fatty acid binding protein seen by molecular dynamics computer calculations. *Biophys. J.* **2000**, *78*, 608–25.

(35) García-Moreno E., B.; Dwyer, J. J.; Gittis, A. G.; Lattman, E. E.; Spencer, D. S.; Stites, W. E. Experimental measurement of the effective dielectric in the hydrophobic core of a protein. *Biophys. Chem.* **1997**, *64*, 211–224.

(36) Dwyer, J. J.; Gittis, A. G.; Karp, D. A.; Lattman, E. E.; Spencer, D. S.; Stites, W. E.; García-Moreno E., B. High apparent dielectric constants in the interior of a protein reflect water penetration. *Biophys. J.* **2000**, *79*, 1610–1620.

(37) Nguyen, D. M.; Reynald, R. L.; Gittis, A. G.; Lattman, E. E. X-ray and thermodynamic studies of staphylococcal nuclease variants I92E and I92K: Insights into polarity of the protein interior. *J. Mol. Biol.* **2004**, *341*, 565–574.

(38) Denisov, V. P.; Schlessman, J. L.; García-Moreno E., B.; Halle, B. Stabilization of internal charges in a protein: Water penetration or conformational change. *Biophys. J.* **2004**, *87*, 3982–94.

(39) Schlessman, J. L.; Abe, C.; Gittis, A.; Karp, D. A.; Dolan, M. A.; García-Moreno E., B. Crystallographic study of hydration of an internal cavity in engineered proteins with buried polar or ionizable groups. *Biophys. J.* **2007**, *94* (8), 3208–16.

(40) Damjanović, A.; García-Moreno E., B.; Lattman, E. E.; García, A. E. Molecular dynamics study of water penetration in staphylococcal nuclease. *Proteins: Struct., Func., Gen.* **2005**, *60* (3), 433–49.

(41) van Gunsteren, W. F.; Dolenc, J.; Mark, A. E. Molecular simulations as an aid to experimentalists. *Curr. Op. Struct. Biol.* **2008**, *18*, 149–153.

(42) Mu, Y. G.; Kosov, D. S.; Stock, G. Conformational dynamics of trialanine in water. 2. comparison of amber, charmm, gromos and opls force fields to nmr and infrared experiments. *J. Phys. Chem. B* **2003**, *107*, 5064–5073.

(43) MacKerell, A. D., Jr.; Bashford, D.; Bellott, M.; Dunbrack, R. L., Jr.; Evanseck, J.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph, D.; Kuchnir; L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, I. W. E.; Roux, B.; Schlenkrich, M.; Smith, J.; Stote, R.; Straub, J.; Watanabe, M.; Wiorkiewicz-Kuczera, J.; Yin, D.; Karplus, M. All-hydrogen empirical potential for molecular modeling and dynamics studies of proteins using the CHARMM22 force field. *J. Phys. Chem. B* **1998**, *102*, 3586–3616.

CI800263C