CHARMM on the Open Science Grid

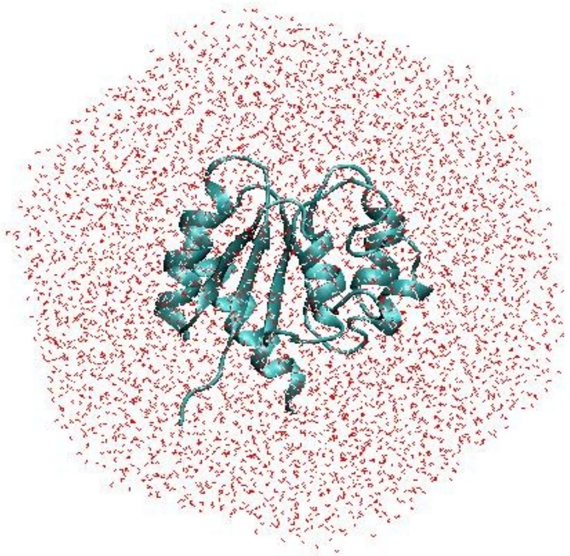
Tim Miller

Laboratory of Computational Biology

U.S. Department of Health and Human Services
National Institutes of Health
National Heart, Lung, and Blood Institute

Molecular Simulation and CHARMM

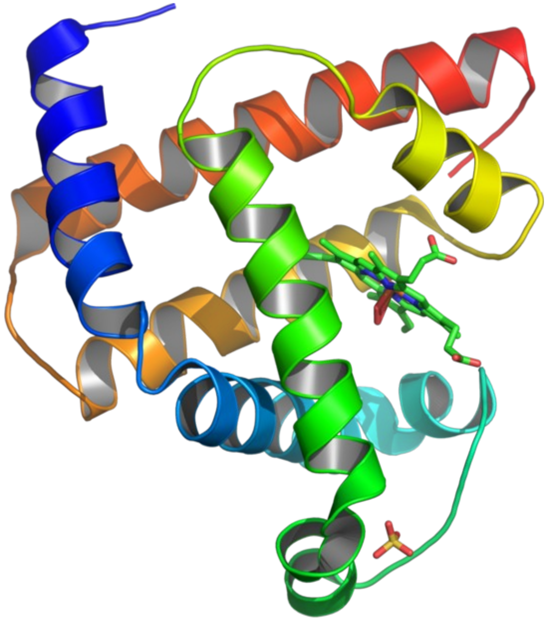
- CHARMM (Chemistry at Harvard Macromolecular Mechanics) is a widely used program for modeling, simulation, and analysis of biological macromolecules.
- The widest use of the program has been for molecular dynamics (MD) simulations.



- In "classical" MD, atoms (including those in solvent) are described explicitly by a **force field** (CHARMM27) that gives the energy of bonds, angles, dihedral angles, van der Waals forces, etc.
- The force field is used to numerically integrate over Newton's second law of motion to produce updated positions of the atoms.

Issues with Molecular Dynamics

- On a single core of a 2.33 GHz Intel Clovertown it takes almost 10 minutes to simulate 1 picosecond of a moderate sized system.
- At this rate it would take nearly 20 years to run dynamics for 1 μs. And this is without using quantum mechanics based methods.



- At optimal scaling, we can reduce the time by around a factor of 8, which brings us down to 2.5 years.
- Many biologically important processes such as protein folding enzyme catalysis, and conformational rearrangement take place on 1 μs time scales.

More on Molecular Dynamics

- A key goal of an MD simulation is to get a statistical sample over the entire conformation space, but because simulation is so expensive computationally we often do not get this sampling.
- The Langevin equation is often used to maintain a constant temperature throughout the simulation; this is called Langevin dynamics (LD).
- Self-guiding can be added to simulations to enhance the efficiency of conformational searching.
- Self-guided Langevin Dynamics (SGLD) is a relatively new method that combines the LD with self-guiding.

Why use the OSG?

Many biological processes are statistical in nature.

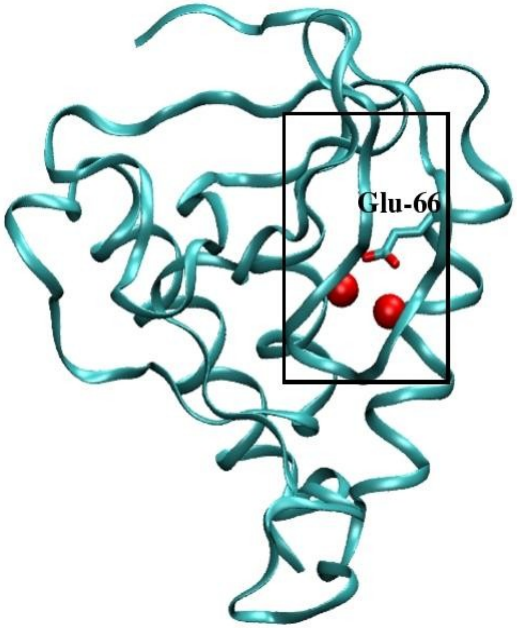
- Simulations of such processes parallelize naturally because many independent jobs must be run.

Performance of multiple simulations on OSG will help with:

- Accuracy
  - Sampling necessary for calculation of experimental observables
  - Simulate mutants, compare their behaviour and compare with experiments
  - Test several force-fields - different force-fields may produce different results
- Describing long timescale processes (μs and longer)
  - Run large number of simulations (increase probability of observation of important events)
  - Test different "alternative" methods

Example problem: hydration of the interior of Staphylococcal Nuclease

The presence of water in the interior of a protein can affect its role in various biological processes.

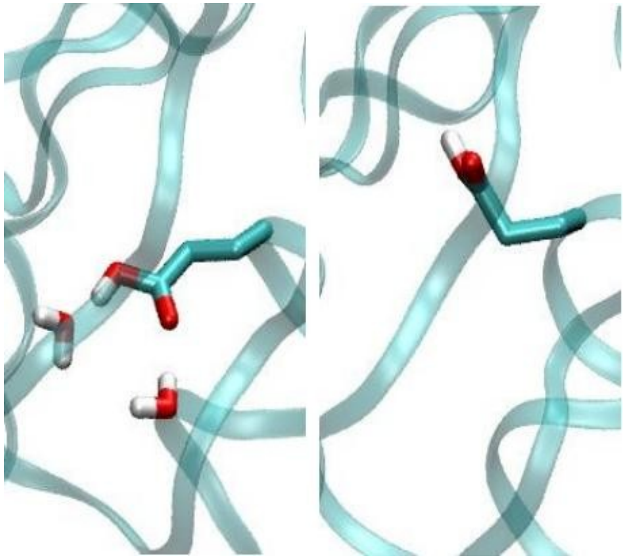


Staphylococcal Nuclease (SN) is an enzyme that cleaves phosphodiester bonds in nucleic acids.

For SN, structure data obtained at different temperatures disagrees on the presence and number of waters at the Glu-66 sidechain. LD simulations can help solve this puzzle.

Hydration of SN (continued)

- Two different conformations the Glu-66 sidechain have been found by simulation, straight and twisted.
- When the sidechain appears straight and extended, 1 or 2 waters are associated with it. When it is twisted back into the protein core, there are no waters.
- The straight conformation occurs most often in simulation, and it is the only one detected by crystallographers.

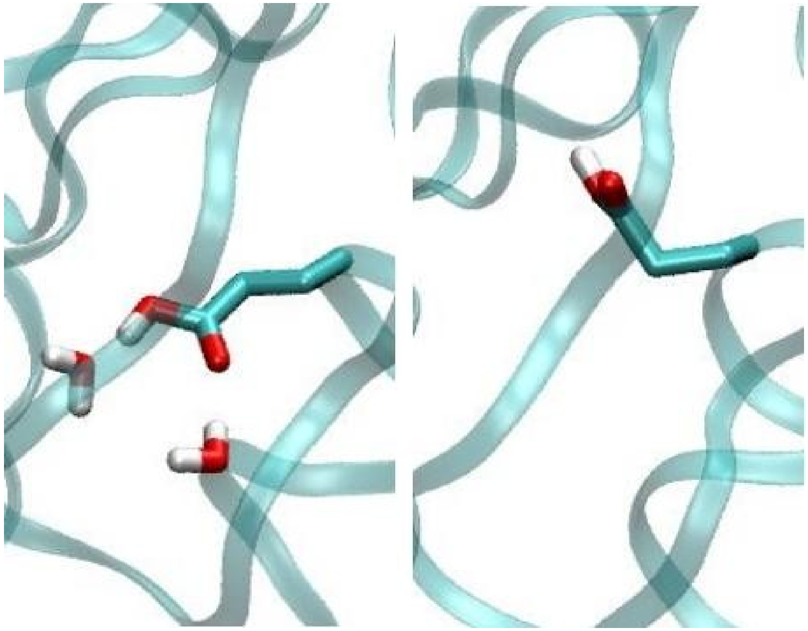


- A key question is how can we know exactly if and how many waters are present.

Our approach to the problem

Previously we performed 10 x 10 ns long simulations, varying the initial velocities of each. This did not yield sufficient statistics to determine the populations of the two conformations!

Since observed conformations seem to depend on initial velocities, we decided to run a very large number of shorter (2 ns) simulations.



We wanted to vary initial velocities, so for each conformation we ran 40 x 2 ns dynamics simulations, for a total computing time of around 42,000 CPU hours!

What we wanted to test

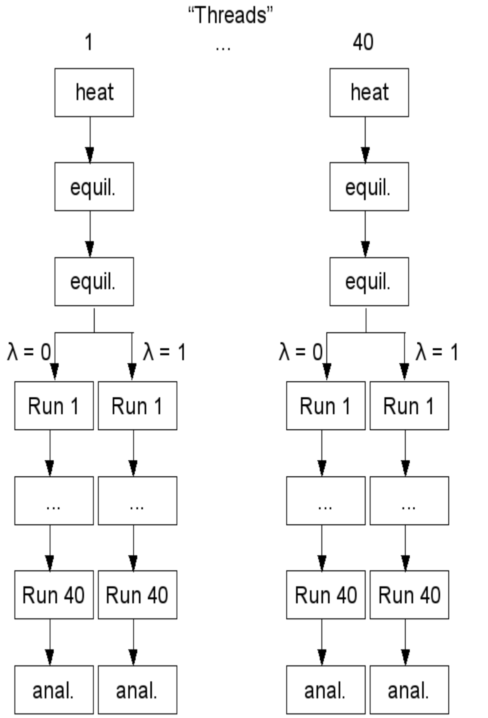
- Initial conditions: We wanted to see if the results made a difference if we started the structures dehydrated as opposed to hydrated.
- Differences between LD and SGLD: SGLD should in theory give more “flips” between conformations since it samples more efficiently.
- For each of the forty simulations per structure, we ran both LD and SGLD as shown below and observed both the number of waters associated with Glu-66 and the conformation of the sidechain.

Number of 2 ns simulations

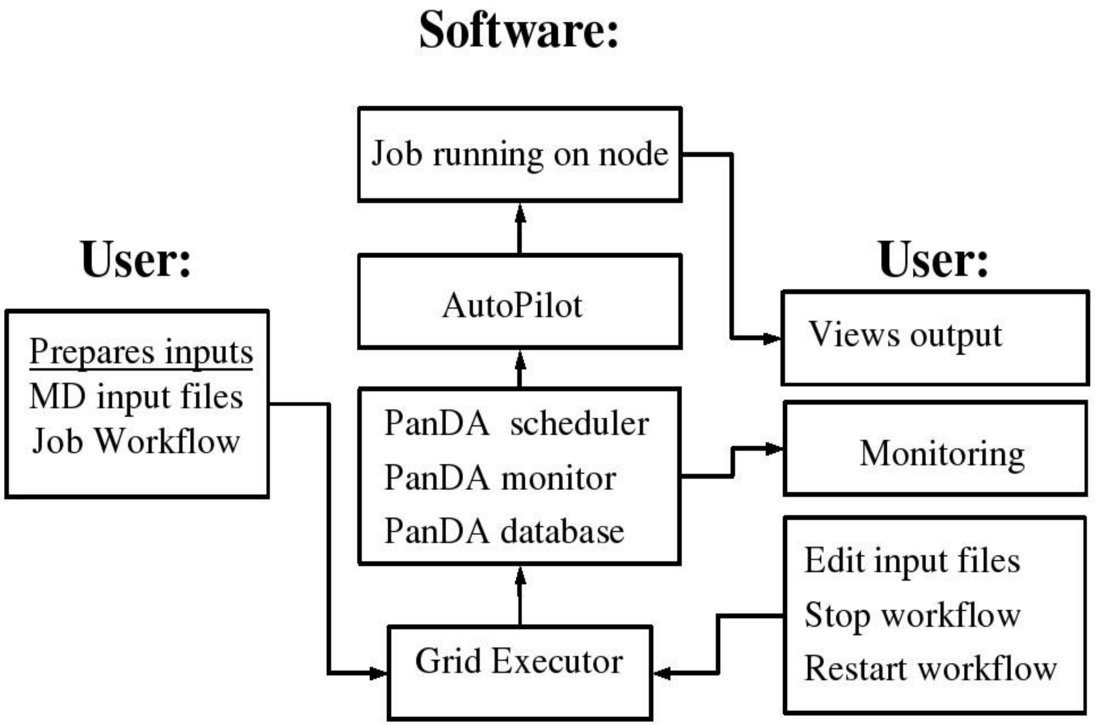
Hydrated		Dehydrated	
LD	SGLD	LD	SGLD
40	40	40	40

Description of the Workflow

- The initial structures were first heated for 100,000 steps (100 ps).
- Then they were equilibrated for 100 ps.
- The two nanoseconds (2 million steps) of dynamics were run.
- The number of waters and the dihedral angle of the Glu-66 residue was extracted from the trajectory and statistics were collected.
- Due to runtime length restrictions at OSG sites, we decided to keep each individual grid job to 50,000 steps (which takes about 12 hours to run on a typical grid node). The image to the right shows two “threads” with many “waves” (individual jobs).



Software set up



We need to run a lot of different jobs in order, i.e. a workflow. We use our own software to keep track of the jobs and PanDA/AutoPilot for submission and monitoring.

What the user must do

1. Create CHARMM input scripts for all the various types of jobs to be run.
2. Prepare the structure files and other needed inputs to CHARMM (e.g. topology and parameter files).
3. Package all of these up in a tarball.
4. Write out the workflow and edit shell scripts to point to correct data repositories.
5. Start the executor program.
6. Watch the result files roll in and keep an eye on the logs for problems.
7. If necessary, edit and restart the workflow,

Example Workflow Script

```
JOB heat USE heat.inp
JOB heat2 USE heat2.inp
JOB eq USE eq.inp
JOB md USE run.inp
JOB analp USE anal_phipsi.inp
JOB analw USE anal_water.inp

# section 2: threads
NTHREADS 30
THREADPARAMS 'I=[threadid]'

# section 3: Default input
requirement and output production
REQDEFAULT [PREVWAVE .res]
PRODEFAULT [.res,.trj,.pdb,.log]

# section 3: order
# The ONLY keywords are
# BRANCH ... REJOIN
# TEST ... ENDTEST
# LABEL, RESULT, ELSE
BEGIN
```

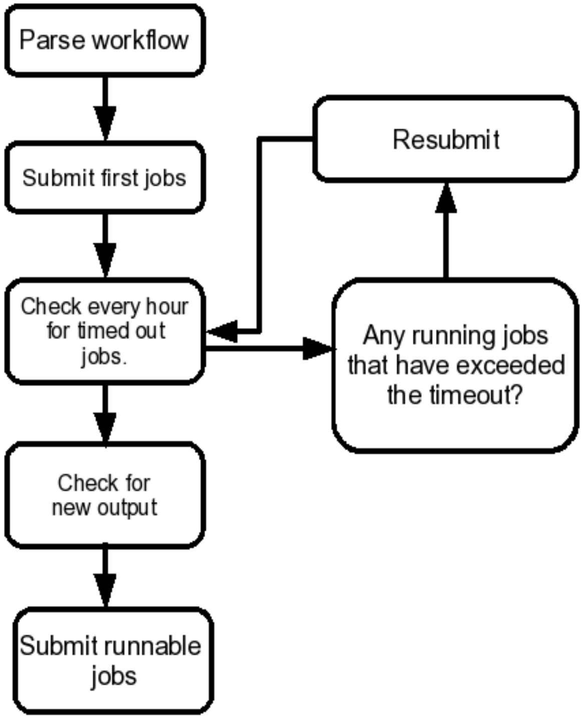
```
heat REQUIRES [NONE NONE]
heat2
eq*2
BRANCH A APPENDPARAMS G=0
  md*20
  analp REQUIRES [ALLDONE
.res,.trj] PRODUCES
[.dih1,.dih2]
  analw REQUIRES [ALLDONE
.res,.trj] PRODUCES [.wat]
BRANCH B APPENDPARAMS G=1
  md*20
  analp REQUIRES [ALLDONE
.res,.trj] PRODUCES
[.dih1,.dih2]
  analw REQUIRES [ALLDONE
.res,.trj] PRODUCES [.wat]
END
```

Our own grid executor!

We designed and built software called a grid executor to submit and manage jobs on our behalf.

It is written in Python and runs as a daemon, submitting jobs and checking for their output.

If a job does not return a result within a set period of time, it is resubmitted.


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

The executor was inspired by DAGMAN for Condor. Jobs can have multiple dependencies and require different inputs and output. The executor is not in any way tied to CHARMM.

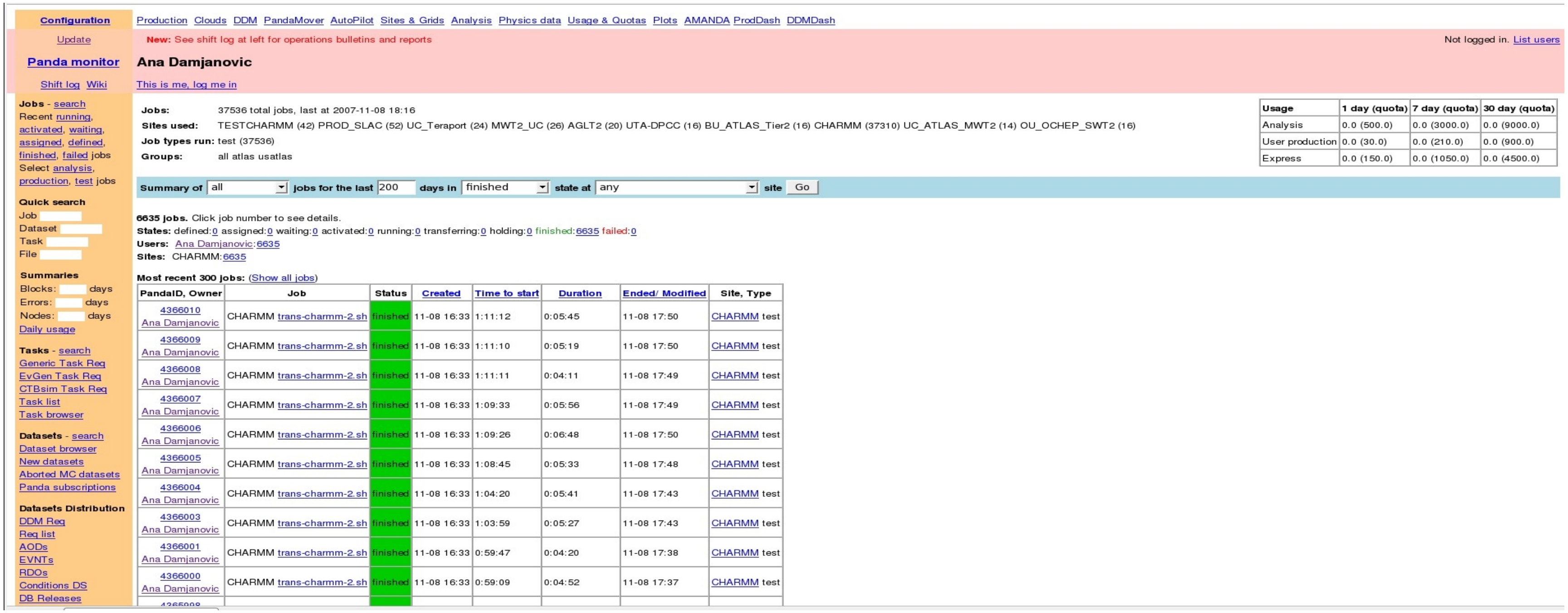
What happens on the grid stays on the grid (or not)

- The grid executor submits jobs when the necessary prerequisite jobs have been run.
- The grid jobs are custom written Python scripts, which are responsible for the following tasks:

1. Downloading a tarball containing the CHARMM binary and necessary input scripts and support files.
2. Getting a list of prerequisites and obtaining them from the data store.
3. Running CHARMM with the proper command line arguments and input script.
4. Making sure CHARMM completes OK and then sending back the results.

Job submission and monitoring

We use PanDA for job submission and monitoring. This goes us nice management tools and a Python API for defining jobs. Also, PanDA is integrated with AutoPilot which ensures (as much as possible) that remote nodes are able to run the job before it is started.



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

Ana Damjanovic

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) CHARMM (37310) UC_ATLAS_MWT2 (14) OU_OCHEP_SWT2 (16)
Job types run: test (37536)
Groups: all atlas usatlas

Usage	1 day (quota)	7 day (quota)	30 day (quota)
Analysis	0.0 (500.0)	0.0 (3000.0)	0.0 (9000.0)
User production	0.0 (30.0)	0.0 (210.0)	0.0 (900.0)
Express	0.0 (150.0)	0.0 (1050.0)	0.0 (4500.0)

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

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

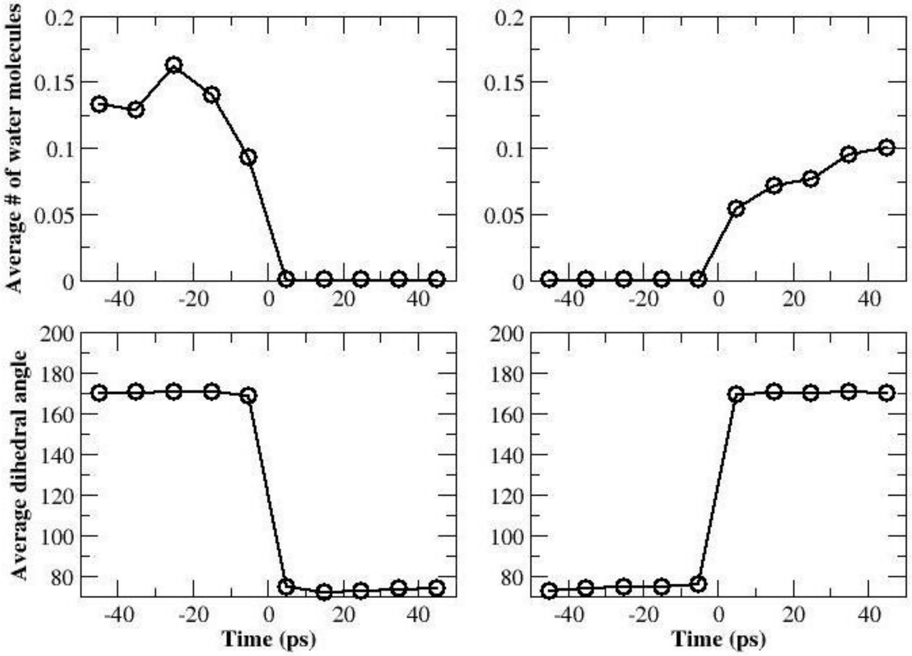
Most recent 300 jobs: (Show all jobs)

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
4365998							

Basic Results of Simulations

- We observed more straight than twisted conformations, which agrees with previous simulations.
- The two systems are still equilibrating, i.e. they have not settled into their final stable conformations. Therefore the results given are qualitative, not quantitative.
- However, the results appear to be converging (i.e. the effect of the initial hydration state becomes less significant over time).
- We confirmed that the straight conformation is more hydrated than the twisted one.
- We found that during the 2nd ns, water molecules can become associated with the Glu-66 sidechain even when it is in the twisted state.

Effect of Dehydration



- The initial hydration state dramatically affects the results.
- The state that is initially hydrated prefers the straight conformation ~ 20% more frequently than the initially dehydrated state.

- The conformational switch from straight into twisted often occurs when the straight sidechain is dehydrated.
- Flips from straight into twisted occur after a drop in hydration (as shown in the figure).
- When structures flip from twisted to straight, average number of waters present increases more gradually.

Differences between LD and SGLD

- Prior work has shown that with Self-guided Langevin Dynamics (SGLD) the amount of protein helix and strand fraying is about twice as large as with regular Langevin Dynamics (LD). Remember, SGLD samples configurations more efficiently!
- In this study, however, the amount of “flips” between the two structures was only ~ 16% greater with SGLD.
- SGLD does appear to favor the curled conformation more (by about 7%) and the dehydrated state (by ~ 8%).
- It appears that LD and SGLD simulations are converging in the same direction, but longer simulations are needed to verify this.
- This work will be published in a forthcoming paper by Damjanovic *et al.*

Summary

- CHARMM is a complex and full featured program for molecular simulation.
- Through our own custom written software, we have made CHARMM run on the Open Science Grid.
- This procedure is not specific to CHARMM – other software such as NAMD, GROMACS, AMBER, etc. can use the grid executor.
- We used our infrastructure to explore the effect internal hydration has on the conformation of Staphylococcal Nuclease.
- The simulations showed that the internal hydration does appear to effect the conformation of the Glu-66 sidechain. They also showed that the initial condition of the hydration is significant.
- More samples need to be taken before any quantitative conclusions can be drawn.

Future directions

- We are continuing the simulations. A third nanosecond is in progress. We hope to see further evidence of convergence.
- We need to be able to run for longer time periods. For that reason, we need to be able to use MPI. A test MPI PanDA queue is in operation, and we are just waiting for some authentication issues to be resolved.
- We want to bring the benefits of grid computing to the wider molecular simulation community. To that end, we are in the process of organizing a new OSG VO, **biomolsim**, as a central point for researchers interested in using the grid for biomolecular simulations.

Acknowledgements

Collaborators:

- Ana Damjanovic (NHLBI, Johns Hopkins University)
- Petar Maksimovic (JHU)
- Torre Wenaus (Brookhaven National Laboratory)
- Bertrand Garcia Moreno E. (JHU)
- Bernard R. Brooks (NHLBI)

Open Science Grid:

- Ruth Pordes
- Frank Wuerthwein
- Alain Roy