SOP-GPU
Examples

WW Domain

Artem Zhmurov
(zhmurov@gmail.com)

Valeri Barsegov
(Valeri_Barsegov@uml.edu)

University of Massachusetts at Lowell

November 24, 2010

1 System

The WW domain, which consists of N=34 residues (Protein DataBank (PDB) code: 1PIN) [5]), has been extensively studied experimentally [3, 2] and computationally [4, 1]. The WW domain involved in the regulation of several cellular processes, and it has been a paradigm for describing folding and unfolding of the β-sheet proteins. The mechanical unraveling of the WW domain can be described by the single-step kinetics of unfolding, $F \rightarrow U$, from the folded state F to the unfolded state U. The linear tandems $(WW)_4$ and $(WW)_6$ of 4 and 6 domains WW can be constructed by connecting the N- and C-termini of the adjacent WW domains using linkers of neutral residues, which will be done in this example using the "sop-top" utility built into the SOP-GPU package. Short flexible linkers do not change the micromechanics of unfolding in multimeric proteins, such as the multimers $(WW)_4$ and $(WW)_6$ described in this example. Since WW is a small protein, here we will exemplify the use of the many-runs-per-GPU approach, which allows one to run several independent trajectories on a single GPU device simultaneously. In general, the number of independent simulation runs (M) should be adjusted so that the total number of residues, i.e. $M \times N = 34 \times M$ in this example, is $\sim 10-50$ times larger than the number of Arithmetic Logic Units (ALUs) on the GPU. The total number of residues is $M \times (4N + 5L)$ for the fragment $(WW)_4$ and $M \times (6N + 7L)$ for the fragment $(WW)_6$, where L=4 is the linker length.

2 This example

In this example we have included four folders: folder "WW" is for the simulations of forced unfolding of the WW monomer, folder "WW_1" is for the WW monomer with linkers added to its terminal ends, folder "WW_4" is for the $(WW)_4$ fragment, and folder "WW_6" is for the fragment $(WW)_6$. Each set of simulations starts from the file "ww.pdb", which is the all-atom representation of the WW domain, extracted from the 1PIN.pdb file from the Protein Data Bank. Below, is a short description about how to set up simulations for the WW monomer (folder "WW"). The simulation procedure for other systems ($(WW)_4$ and $(WW)_6$) is similar; all you need to do is "cd" to the folder with a particular example and type the following simple commands in the terminal window:

```
$ sop-top ww_top.sop
$ sop-gpu ww_equil.sop
$ sop-gpu ww_pull.sop
```

3 Coarse-graining and topology description

The topology file provides the description of all the C_α-particles in the system, i.e. how they are bonded (chain connectivity) and what pairs of amino acid residues form the native contact (secondary structure, native folded state). Since the molecular topology is derived from the all-atomic structure of the protein in question, the topology is specified at the so called coarse-graining stage, when a simplified C_α-based model is created from the all-atomic (PDB) structure. In the simulation folder, you should find the configuration file for creating

topology ("ww_top.sop"). This configuration file is a set of pairs of parameter values, which are used in the simulations (similra to NAMD or Gromacs configuration file). When coarse-graining the system of interest, the input for this program is the initial pdb file and the output are the two files - one for topology and the other for the coarse-grained coordinates. Also, the configuration file contains parameters for topology definition. These include C_α to C_α and side-chain to side-chain cut-off distances, specified by parameters R_limit_bond and SC_limit_bond, respectively, as well as the value for ε_h (eh). These are the most important parameters in the SOP model. The first two parameter specify whether a specific pair of amino acids form a native contact (in the native state), whereas the later specifies the average strength of native contacts. In case of the WW domain, we used ε_h=1.5$kcal/mol$ and so called "FullGo" definition for the cut-off distances. For more complex systems, these parameter have to be estimated based on the results of all-atom Molecular Dynamics (MD) simulations. Alternatively, these parameters can be adjusted so that the results of pulling simulations agree with the experimental force spectra. To create the molecular topology, using the "sop-top" utility in SOP-GPU, run the following command in terminal window:

```
$ sop-top ww_top.sop
```

This will save both the topology file and the pdb structure file in "topologies" and "structures" folders, respectively.

4 Running simulations

Each simulation run is done in two steps. The first, equilibration step is done to obtain proper randomization of the initial conformation for all the trajectories. The second step is the actual pulling simulation run. In this example, there are two configuration files provided for each system: one file ends with the "equil" suffix, and the other file has "pull" suffix. These files specify all the simulation parameters, including the temperature, the total number of steps of a simulation run, the seed for the (pseudo)-random numbers generator, etc. Also, as the input files, the filenames for system structure and topology are specified; in the output, there are names for the .dcd coordinates file, energy .dat file, and restart and final coordinates. Since the configuration files will start 100 simulation runs on one GPU device (many-runs-per-GPU), this number is specified by the "runnum" parameter in the configuration file. In order to run simulations, the name of this parameters file should be passed as the first argument to the SOP-GPU program. Hence, to start equilibrium simulations, one should type:

```
$ sop-gpu ww_1-100_equil.sop
```

After the system has been equilibrated in all the independent simulation runs, the pulling simulations can be started by using the following command:

```
$ sop-gpu ww_1-100_pull.sop
```

Alternatively, one can use the shell script provided in order to start the second, pulling step after the first equilibrium step. This can be achieved by typing the following command:

```
$ sh ww_1.sh > ww_1.out &
```

For further details about the configuration files and the simulation output, please refer to the configuration files and to the SOP-GPU manual in the Documentation section.

5 Results and analysis

To get the force-extension curves, plot the results in column 4 against the results in column 2 in the pulling output .dat file (pull/ww_1.dat). Note that the unfolding force is reported in units of $kcal/mol$Å, and the distance is given in units of Å. Check Ref. [6] for more information about the simulation data analysis.

References

[1] M. S. Cheung, D. Klimov, and D. Thirumalai. Molecular crowding enhances native state stability and refolding rates of globular proteins. *Proc. Natl. Acad. Sci. USA*, 102:4753 – 4758, 2005.

[2] N. Ferguson, J. Berriman, M. Petrovich, T. D. Sharpe, J. T. Finch, and A. R. Fersht. Rapid amyloid fiber formation from the fast-folding WW domain FBP28. *Proc. Natl. Acad. Sci. USA*, 100:9814 – 9819, 2003.

[3] Marcus Jger, Houbi Nguyen, Jason C Crane, Jeffery W Kelly, and Martin Gruebele. The folding mechanism of a β-sheet: the WW domain. *J. Mol. Biol.*, 311:373 – 393, 2001.

[4] J. Karanicolas and C. L. Brooks. The structural basis for biphasic kinetics in the folding of the WW domain from a formin-binding protein: Lessons for protein design? *Proc. Natl. Acad. Sci. USA*, 100:3954 – 3959, 2003.

[5] E. K. Koepf, H. M. Petrassi, M. Sudol, and J. W. Kelly. WW: An isolated three-stranded antiparallel β-sheet domain that unfolds and refolds reversibly; evidence for a structured hydrophobic cluster in urea and gdnhcl and a disordered thermal unfolded state. *Protein Sci.*, 8:841 – 853, 1999.

[6] A. Zhmurov, R.I. Dima, and V. Barsegov. Order statistics theory of unfolding of multimeric proteins. *Biophys. J.*, 99:1959 – 1968, 2010.