



ProDy

Protein Dynamics & Sequence Analysis

coMD Tutorial

Release 1.7

Cihan Kaya

November 13, 2015

CONTENTS

Learn global transition pathways with multiscale hybrid methodology, collective molecular dynamics.

INTRODUCTION

Collective Molecular Dynamics is a [VMD¹](#) plugin GUI and a Python module developed for setup and analysis of simulations described in [\[MG13\]](#) (page ??).

1.1 Installation

1. [VMD²](#) 1.9.2 or later is required for using GUI. '[NAMD](#)' is required for running molecular dynamics simulations and latest version of [ProDy³](#) is required for elastic network model calculations. Following are required for performing coMD calculations:
 - [Python⁴](#) 2.7
 - [NumPy⁵](#) 1.3 or later
2. Download one of the following archive files:
 - `comd_plugin_files.tgz`
 - `comd_plugin_files.zip`
3. Extract contents of the archive and copy `comd` folder to VMD TCL plugins directory, i.e. `$VMDDIR/plugins/noarch/tcl/`.

Then, insert following line into `$VMDDIR/scripts/vmd/loadplugins.tcl` at line 200:

```
vmd_install_extension comd comd_tk "Modeling/DruGUI"
```

If you are not sure where VMD directory is located, run `vmd`, and type the following command line in the VMD console:

```
global env; puts $env(VMDDIR)
```

1.2 coMD Plugin

Collective Molecular Dynamics (coMD) plugin, shown below, has a panel to setup, collective molecular dynamics simulations:

- *Simulation Setup* (page ??)

¹<http://www.ks.uiuc.edu/Research/vmd>

²<http://www.ks.uiuc.edu/Research/vmd>

³<http://prody.csb.pitt.edu>

⁴<http://www.python.org>

⁵<http://www.numpy.org>

Collective Molecular Dynamics v1.0

Protein structures:

Initial PDB:

Final PDB:

Initial PDB chain ID: Final PDB chain ID:

Ionization parameters

Topology File:

Box padding (A): Add counter ions:

Minimization parameters

Parameter File:

Temperature (K): Minimization length (ps):

ANM-MC-Metropolis options:

ANM cutoff (A): Deviation (A):

TMD options:

Spring constant: TMD length:

Simulation options:

No of coMD cycles: No of physical cores:

Output options:

Output folder:

Output prefix:

The rest of the tutorial will show you how to use this panel, and described required inputs and outputs from analysis steps.

1.3 Tutorial Files

Files in the following archives can be used to follow this tutorial:

- coMD Tutorial Files (TGZ)
- coMD Tutorial Files (ZIP)

Here is a list of these files:

350K	Nov	4	09:46	lake.pdb
303K	Nov	4	09:47	4ake.pdb
2.8M	Nov	4	23:01	final_ionized.pdb
17M	Nov	4	23:01	final_trajectory.dcd
2.2M	Nov	4	23:01	initial_ionized.pdb
13M	Nov	4	23:01	initial_trajectory.dcd

1.4 How to Cite

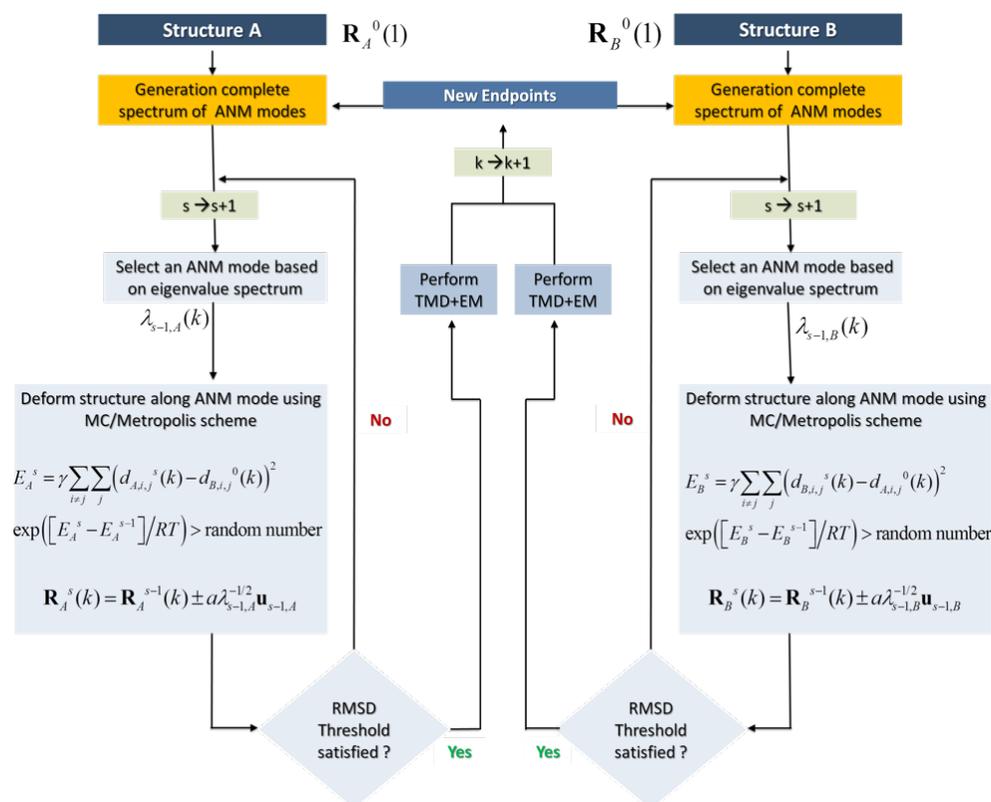
If you benefited from collective molecular dynamics in your research, please cite the following paper:

BACKGROUND

Biological functions of biomolecules is accomplished by conformational motions that enable the molecules to sample functional substates. Some of these substates are short-lived, and therefore difficult (or even impossible) to detect experimentally. Here, computational tools are useful to predict plausible transition pathways, which can be probed experimentally. They may also provide insights into accessible conformational states, or the energy landscape, which often determines the structural changes taking place during interactions with other proteins and ligands. Full atomic MD simulations often fail in sampling large-scale cooperative changes in conformation (global motions) especially in the case of large molecular systems. Coarse-grained approaches like ANM fail in providing atomic details, while they provide an accurate description of global dynamics. We have developed a hybrid method, coMD, that takes advantage of the ability of ANM to sample global modes so as to direct MD simulations. This way we can sample global motions at full atomic resolution. Each step in coMD is a collective movement of the entire molecule along a given ANM mode, selected by a Monte Carlo/Metropolis algorithm.

METHODOLOGY

Below is a schematic description of the methodology for sampling the transition pathway between two known structures A and B. The same approach can be used for sampling the conformational space in the vicinity of both substates, simply by defining the Metropolis criteria to accept all moves, irrespective of the energy change at the last step. Thus, the method can be used for either characterizing the pathway(s) between two endpoints A and B, or mapping the conformational space in general for exploring alternative substates.



In a nutshell, the method consists of the following steps applied to each endpoint in parallel: (i) selecting a given ANM mode using a Monte Carlo algorithm based on the weight (inverse eigenvalue, square root) of each mode, (ii) examining if this mode allows for approaching the other endpoint, and accepting/rejecting it with the help of a Metropolis algorithm based on an energy function that penalizes the increase in RMSD from the endpoint (this condition can be removed to allow for unconstrained sampling), (iii) generate a target conformation based on the selected mode, (iv) use this target conformation to simulate (by TMD) the transition toward this target conformation - which defines an intermediate conformer in the process of

sampling the transition path (or the conformational space), (v) energy minimize, and (vi) go back to step (i) by redefining the intermediate conformation as the new endpoint, until completion of the sampling (e.g. if the two endpoints generated in parallel are sufficiently close to each other).

SIMULATION SETUP

System that contains protein structures with water and counter ions in collective molecular dynamics simulations can be prepared using the following interface:

4.1 Input Files

.pdb files for both initial and final structure of the protein are required from the user. You can learn how to prepare these files from [NAMMD tutorials](#)¹.

4.2 Protein Structures

1. First, select initial .pdb (structure) files and final .pdb (structure) files. You can use the adelynate kinase files provided for this tutorial. (1ake.pdb, 4ake.pdb) Alternatively, structure and coordinate files for a protein of interest can also be used. These files should contain all atoms required for protein stability and function, these may include cofactors and metal atoms. Crystallographic water molecules may also be retained in the structure.
2. Select the corresponding chain ID's for initial and final structure. For adelynate kinase both chain ID's should be 'A'.

4.3 Ionization Parameters

1. To build the topology of the structure, a topology file needs to be provided. This file will parametrize the topology of structures e.g. bond lengths and bond angles. If not given, it will use CHARMM22 all-hydrogen topology file for proteins and lipids. For more information visit [Topology](#)² pages.
2. Enter simulation box padding (distance from the protein to box surfaces) and select whether you would like to add counter ions. Collective Molecular Dynamics GUI uses [Solvate](#)³ and [Autoionize](#)⁴ plugins to add water and ions.

¹<http://www.ks.uiuc.edu/Training/Tutorials/>

²http://www.ks.uiuc.edu/Research/vmd/plugins/membrane/top_all27_prot_lipid.inp

³<http://www.ks.uiuc.edu/Research/vmd/plugins/solvate/>

⁴<http://www.ks.uiuc.edu/Research/vmd/plugins/autoionize/>

Collective Molecular Dynamics v1.0

Protein structures:

Initial PDB:

Final PDB:

Initial PDB chain ID: Final PDB chain ID:

Ionization parameters

Topology File:

Box padding (A): Add counter ions:

Minimization parameters

Parameter File:

Temperature (K): Minimization length (ps):

ANM-MC-Metropolis options:

ANM cutoff (A): Deviation (A):

TMD options:

Spring constant: TMD length:

Simulation options:

No of coMD cycles: No of physical cores:

Output options:

Output folder:

Output prefix:

4.4 Minimization Parameters

1. For every cycle, the structures (protein and water) will be minimized by using NAMD. For parametrization of molecular dynamics simulation in NAMD, an input file is required. If not given, it will use Combined CHARMM22 all-hydrogen parameter file for CHARMM22 proteins and CHARMM27 lipids. For more information visit [Parameter](#)⁵ pages.
2. For minimization, two additional parameters are required. The first one is the temperature that the minimization will be performed and the second one is the total number of steps to perform minimization. Default parameters are 310 K and 50 steps.

4.5 Conformational Sampling with ANM Modes

For sampling of conformations with ANM modes, two parameters are required. ANM cutoff is the parameter to determine the contacts on the proteins for ANM calculations. The second parameter is the deviation parameter. For each step in conformational sampling there is a scale factor for perturbing structure on the direction of ANM mode.

4.6 Targeted Molecular Dynamics Parameters

For every cycle, after conformational sampling with ANM modes toward to final structure, the side chains of the protein needs to move by performing a fast targeted molecular dynamics simulation (TMD). In TMD simulations, two parameters are required. The first one is the spring constant (kcal/mol/A²) and the second one is the length of TMD simulation (ps). For further information visit [TMD](#)⁶ pages.

4.7 Simulation Options

coMD simulations will run in cycles and the number of cycles is the global parameter for simulations. The simulation will run until the number of cycles achieved or the RMSD between initial and final structure is less than 1.5 Å or the reduction in the RMSD is less than 0.15 Å. The other parameter in this block is the number of physical cores on your machine for parallel NAMD simulations.

4.8 Output Options

Select output folder, file prefix to distinguish different runs. Performing multiple simulations to see whether results are reproducible is always a good idea.

4.9 Output Files

There will be an output folder with a python script (py), a tcl script (tcl), two pdb and psf files and two parameter files will be prepared. The tcl script should run from vmd.

For a summary of contents of the final system, see `prefix.log` file.

⁵http://www.ks.uiuc.edu/Training/Tutorials/science/nanotubes/files/par_all27_prot_lipid.prm

⁶<http://www.ks.uiuc.edu/Research/namd/2.10b1/ug/node47.html>

4.10 Simulation

Now you need to run coMD simulations. To perform those simulations it is possible to use the following command:

```
vmd -dispdev text -e your_output_prefix.tcl
```

If you want to run this simulation on cluster, copy this directory on cluster and put following command on file queue system.

When simulations are complete, you can continue with following analysis steps.

COLLECTIVE MOLECULAR DYNAMICS ANALYSIS

After running coMD simulation, the results will be prepared in the same folder named in setup section. You need to download the following files from that folder:

1. `initial_ionized.pdb`
2. `final_ionized.pdb`
3. `initial_trajectory.pdb`
4. `final_trajectory.pdb`

5.1 Preparing Trajectory Files

We recommend that you will follow this tutorial by typing commands in an IPython session, e.g.:

```
$ ipython
```

or with pylab environment:

```
$ ipython --pylab
```

First, we will make necessary imports from ProDy and Matplotlib packages.

```
In [1]: from prody import *
In [2]: from pylab import *
In [3]: ion()
```

coMD simulations will create two different trajectory and we need to use those two trajectories to analyze our simulations. However, the simulation boxes have different number of atoms due to solvation and ionization procedure in the beginning of simulations. We need to load trajectories and related structure files.

```
In [4]: dcd1 = Trajectory('initial_trajectory.dcd')
In [5]: dcd2 = Trajectory('final_trajectory.dcd')
In [6]: dcd1
Out [6]: <Trajectory: initial_trajectory (next 0 of 40 frames; 29031 atoms)>
In [7]: dcd2
Out [7]: <Trajectory: final_trajectory (next 0 of 40 frames; 36588 atoms)>
```

```
In [8]: structure1 = parsePDB('initial_ionized.pdb')
In [9]: structure2 = parsePDB('final_ionized.pdb')
In [10]: structure1
Out[10]: <AtomGroup: initial_ionized (29031 atoms)>
In [11]: structure2
Out[11]: <AtomGroup: final_ionized (36588 atoms)>
```

It is required to link the structure and atoms however, the size of initial and final structure may be different in terms of total atoms. However, the total number of C-alpha atoms must be equal. The structure and trajectory is linked with following commands:

```
In [12]: dcd1.setCoords(structure1)
In [13]: dcd1.setAtoms(structure1.calpha)
In [14]: dcd1
Out[14]: <Trajectory: initial_trajectory (next 0 of 40 frames; selected 214 of 29031 atoms)>
In [15]: dcd2.setCoords(structure2)
In [16]: dcd2.setAtoms(structure2.calpha)
In [17]: dcd2
Out[17]: <Trajectory: final_trajectory (next 0 of 40 frames; selected 214 of 36588 atoms)>
```

It is not available to concatenate trajectories inside Python (It is available for ProDy command line application). Therefore, we prepare two trajectory files for only C-alpha atoms with following commands:

```
In [18]: writeDCD('initial_filtered.dcd', dcd1)
Out[18]: 'initial_filtered.dcd'
In [19]: writeDCD('final_filtered.dcd', dcd2)
Out[19]: 'final_filtered.dcd'
```

5.2 Concatenating Trajectory Files

After having two trajectory files, we can start analyzing multiple **:files:.dcd** files. First, the trajectory files will be concatenated.

```
In [20]: traj = Trajectory('initial_filtered.dcd')
In [21]: traj.addFile('final_filtered.dcd')
```

5.3 Principal Component Analysis

For concatenated trajectory we created a PCA object, created covariance matrix and calculated eigenvalues and eigenvectors.

```
In [22]: pca = PCA('Adelynate Kinase coMD')
In [23]: pca.buildCovariance(traj)
```

```
In [24]: pca.calcModes()
```

The first half of the trajectory is for initial structure and the second half of the trajectory is for final structure. Those two trajectories are separated.

```
In [25]: forward = traj[0:40]
```

```
In [26]: backward = traj[40:]
```

Before visualizing the trajectories on principal components, it is required to translate those structures on top by using `superpose()` function.

```
In [27]: forward.superpose()
```

```
In [28]: backward.superpose()
```

5.4 Visualization of Trajectories

Finally, the trajectories can be plotted by using `showProjection()` function:

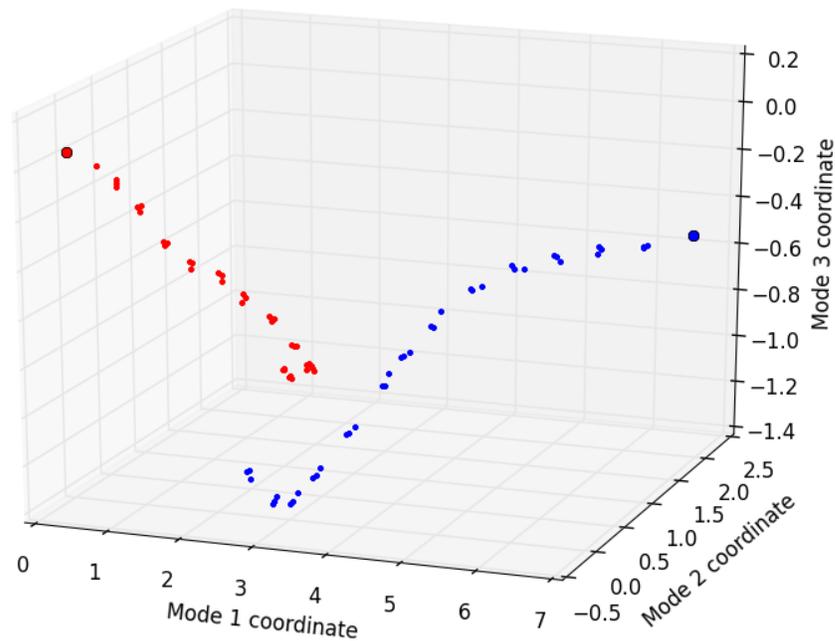
```
In [29]: showProjection(forward, pca[:3], color='red', marker='.');
```

```
In [30]: showProjection(backward, pca[:3], color='blue', marker='.');
```

```
In [31]: showProjection(forward[0], pca[:3], color='red', marker='o');
```

```
In [32]: showProjection(backward[0], pca[:3], color='blue', marker='o');
```

The plots will be in the following form:



Now we calculated the modes and we can write them to a .nmd file for viewing in normal mode wizard.

```
In [33]: writeNMD('ake_pca.nmd', pca[:3], structure1.select('calpha'))  
Out [33]: 'ake_pca.nmd'
```

LICENSE

Copyright (c) 2013, University of Pittsburgh
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Acknowledgments

Continued development of Protein Dynamics Software *ProDy* is supported by NIH through R01 GM099738 award. Development of this tutorial is supported by NIH funded Biomedical Technology and Research Center (BTRC) on *High Performance Computing for Multiscale Modeling of Biological Systems (MMBios¹)* (P41 GM103712).

¹<http://mmbios.org/>

BIBLIOGRAPHY

- [MG13] Gur M, Madura J, Bahar I Global Transitions of Proteins Explored by a Multiscale Hybrid Methodology: Application to Adenylate Kinase² *Biophys J* **2013** 7:1643–1652

²<http://www.sciencedirect.com/science/article/pii/S000634951300934X>