Examples of Colvars-based protocols:
Association of polyleucine peptides

:

Giacomo Fiorin <giacomo.fiorin@gmail.com>
Temple University, Philadelphia, PA
National Institutes of Health, Bethesda, MD

Jérome Hénin <jerome.henin@ibpc.fr>
CNRS, Paris, France

The latest version of this document is available at:
https://github.com/Colvars/examples/tree/master/15leu2x/Instructions.pdf

The key to the success for a simulation with biases applied to collective variables is to capture the interesting (read: *slow*) degrees of freedom in the system. There are thus two steps:

1. design of collective variables
2. biased simulation protocol

In this exercise, we look at association of two helical polyleucine peptides in vacuum. Association processes are difficult problems because they always involve many degrees of freedom: at least 6 for the association of two rigid bodies, and then internal degrees of freedom of each object may come into play as well. Since we have a limited time for this exercise, we will need to reduce the space to be sampled by restraining some of those degrees of freedom.

The required files for this exercise can be downloaded from:
https://github.com/Colvars/examples (`15leu2x/input` folder)
https://github.com/Colvars/datasets (trajectory files only, `15leu2x/trajectories` folder)

1. Colvar design based on analysis of an unbiased trajectory (VMD)

First we want to characterize the important motions of the dimer and find the potentially interesting regions. Since we don't have a good intuition of the system yet, some initial sampling will be useful. Here we provide a **set of 10 trajectories, labeled `s0` through `s9`**, which correspond to unbiased MD simulations with different starting points, with random placement and orientation of helix B.

Let's analyze these trajectories in terms of different colvars to see how they are distributed, and if there are interesting basins (local free energy minima).

Data from previous simulations

The files are organized as follows:

- `15leu2x.psf`: the system's topology information used by NAMD;
- `15leu2x.s?.pdb`: the starting point of the "`s?`" simulation;
- `15leu2x.s?.equil.dcd`: the trajectory file of the "`s?`" simulation. Note that the first frame in each file represents t = 10 ps, and each simulation is 10 ns long (1000 frames). For your convenience, the backbone of the first helix (VMD selection: `segid HA`) has been aligned to its starting structure, but keep in mind that simulations were carried out without restraints.

Load the trajectories in VMD (files `15leu2x.psf`, `15leu2x.pdb`, and `15leu2x.s?.equil.dcd`). Note that some simulations have the helices in parallel and others in antiparallel orientations.

What difference do you expect that to make?

Calculate and visualize collective variables

To calculate colvar values along the trajectory, we will use the Colvars module built into the latest version of VMD, and its Tcl interface which uses the command *cv*, which is documented here:

http://colvars.github.io/colvars-refman-vmd/colvars-refman-vmd.html

We will enter Tcl commands in Tk Console. Select Extensions->Tk Console from VMD's Main menu. Don't worry if you don't know any Tcl, the few commands you will need are spelled out below, and we provide some helpful scripts (all files with a `.tcl` extension). When you'll need to load an external script file, type in Tk Console:

```
% play <filename>.tcl
```

Don't forget to use Tab-completion if available to enter file and command names faster and without typos!

Once the trajectories are loaded, initialize the Colvars module on VMD's current "top" molecule:

```
% cv molid top
```

then load the configuration file that defines two variables:

```
% cv configfile dist_orient.colvars.in
```

Just type "`cv`" for the built-in help. Experiment with the commands! If you make a mistake, you can always use "`cv reset`" and "`cv configfile`" to reload the colvar definitions.

The first colvar defined here is "`dist`", the distance vector joining the centers of the two helices. That is one way to measure association, with the added detail of *where* exactly around helix A helix B is.

The most basic way to analyze the trajectory is to calculate the values of the collective variables. The **colvar_display** tool will do that interactively for you:

```
% play colvar_display.tcl
% start_colvar_display dist
```

If you animate the trajectory, you can read the real-time value of the variable. When you are done, stop the display with:

```
% stop_colvar_display
```

In this particular example, each helix seen as a rigid body has three rotational degrees of freedom, which can be described by three Euler angles, or more robustly, by a quaternion. That is the quantity returned by the `orientation` colvar component (labeled "`orient`" in our files) which we will use both for analysis and for orientational restraints.

Since quaternion values are not easy to read intuitively, we provide a script to visualize the rotation associated with an orientation colvar in VMD: `rotation_display`. If you have already loaded the file `colvar_display.tcl`, just type:

```
% start_rotation_display orient
```

When you are done with visualization, stop the display with:

```
% stop_rotation_display
```

To access numerical values of the variables currently defined from the trajectory, you can use another script:

```
% play calc_variables.tcl
```

The script is quite short: don't hesitate to have a look at its contents, and customize it to your needs.

If you are already familiar with the Colvars module, you can experiment with different types of variables to analyze these trajectories.

2. Unbinding simulation and free energy estimates (NAMD)

In these simulations, we will *restrain the position and orientation of one helix*, helix A, to make the analysis simpler. In vacuum or in any homogeneous and isotropic environment, this makes no difference. In other situations (e.g. a lipid membrane), the Colvars module provides the option to let helix A free to diffuse and rotate, and model the motion of helix B in the frame of reference of helix A (see documentation for the `fittingGroup` keyword).

Tricks for a quick free energy estimate

We have only a short time to simulate the unbinding process, but we would still like to estimate a rough free energy profile for this process. We have seen that several bound poses (local minima) and pathways are possible, and that they differ largely by the relative orientation of the two helices.

We will resort to several tricks to limit the amount of sampling required:

1. We will try to describe unbinding from one or more of the possible **dimer structures** rather than all local minima; each structure will be treated in a separate simulation. Therefore, we will effectively simulate the reverse process to the equilibration.
2. Instead of trying to sample all pathways, we will restrain the mutual arrangement of the two helices to follow a specific **vector distance** and **relative orientation,** to sample unambiguous pathways.
3. We will apply **non-equilibrium pulling forces**, essentially constant-velocity steered MD, to force the unbinding process to happen quickly; it is very easy to measure the work exerted by the pulling force, however because we are pulling quickly, that work contains some amount of **irreversible work**, and hence provides a relatively poor estimate of the free energy of binding/unbinding.
4. When possible, we will try to obtain a free energy estimate that contains less irreversible work thanks to the **thermodynamic integration** estimator. Besides ABF, this is also available for restraints and for metadynamics biases via the keyword `writeTIPMF`.
   Note: in earlier versions of the software another way of doing this was to enable ABF but *disable the biasing force itself* to calculate the free energy gradient.

Are free energies calculated under orientational restraints relevant?

There are three relevant physical regions to consider: the *dimer state*, the *unbound state*, and the *pathway* linking them.

1. The free energy contribution of the restraint in the dimer state is negligible because we have seen that helices in dimers have little orientational freedom to begin with.
2. The restraint does reduce the rotational entropy of the unbound state, and hence change the free energy of binding. Should it be significant, this change can be calculated by a separate simulation, or even analytically.
3. The unbinding pathway can be affected by the restraint in a complicated way. However, if all we are interested in is the monomer-dimer equilibrium, this bias on the pathway is irrelevant, as the two end-points are well defined and the calculation converges. See “Ideas for further work” regarding how to run a more complete (and longer) calculation.

Finally, note that the orientation coordinate for each helix is orthogonal to the position of the center of the helix, so orientational biases exert no force on the distance between helices.

Input files for your simulations

The required input files for the simulations are:

- `15leu2x.psf`: the same topology file used earlier in VMD. *You do not need to edit this file.*
- `par_all36_prot.prm`: the parameter files for the CHARMM36 force field for proteins (except for the atoms’ partial charges, which are contained in the PSF file). *You do not need to edit this file.*

- 15leu2x.s?.equil.coor: the final structures (in double precision binary format) of the s? simulations, to be used as starting points for your simulations. *You do not need to edit these files (and we dare you try!).*

The main NAMD script is **unbind.namd** and it includes basic simulation parameters as well as configuration options for the Colvars module. *Edit this file to change the number of simulation steps (default: 50,000 steps, or 100 ps) or other simulation parameters.*

For this exercise, the Colvars configuration is included via multiple files, each loaded through cv configfile. Compared to a single file (which is a simpler approach) this allows to easily change the protocol while retaining the same configuration for the collective variables. The files are:

- dist_orient.colvars.in: defines three variables, the vector distance between the centers of mass of the two peptides (labeled dist), the orientation of peptide B (labeled orient), and the Euclidean distance between the two centers of mass (labeled r).
- restrain_helixA.colvars.in: defines restraints on the absolute position of peptide A and its orientation in the laboratory frame. This simplifies the definition of dist and orient, at the price of a small reduction in configurational entropy for peptide A.
- force_helical_structure.colvars.in (optional): defines restraints on two RMSD coordinates that prevent both helices from distorting. Each is restrained towards the same reference coordinates: alpha_ideal_ref.xyz, which contains only alpha carbons.

The actual protocol is defined by smd.namd, where the configuration for a moving harmonic restraint (labeled "drag") is loaded by a cv config command. Loading Colvars configuration directly in the Tcl script processed by NAMD allows using information specific to your current run (e.g. the number of steps is used to define how much should the bias accelerate dynamics).

Run NAMD

The namd2 command should available in the path, and as usual it is good practice to redirect the standard output to a log file:

```
$ namd2 unbind.namd > unbind.log &
```

Analyze results

In addition to loading back the DCD trajectory file in VMD as you did earlier, you can now analyze additional files produced by the Colvars module:

- <jobname>.colvars.traj: a text file that contains the values of the collective variables and related quantities. Lines beginning with "#" contain the variables' names, aligned with the respective columns. (The labels may change when you load a new file).
Tip: you may use the plot_colvars_traj.py script (see syntax below) to print only chosen variables from this file.
- <jobname>.pmf: the free energy profile as a function of the distance "r" calculated by thermodynamic integration. This is written during the simulation, with the same frequency as the restart files. (Note: metadynamics, if activated, would use the same naming scheme).

We recommend comparing the free energy calculated by TI/ABF, reported in <jobname>.pmf, with the non-equilibrium work performed by the "drag" restraint. Although the latter is multidimensional, it is a fair approximation to plot the work performed by the restraint "W_drag" against the value of "r" over time:

```
$ ./plot_colvars_traj.py --output smd-work.dat –-variables r W_drag --
        <jobname>.colvars.traj
$ ./plot_colvars_traj.py --plot smd-work.pdf –-variables W_drag --plot-x-axis r
        <jobname>.colvars.traj
```

How do the free energies calculated by these two methods compare? Keep in mind that they come from the same simulation.

Set up additional simulations

You can now edit unbind.namd to perform another simulation, e.g. starting from a point other than "s0" or using a different protocol.

If you choose a different starting point, edit the value of starting_structure in unbind.namd, or set it as an environment variable: the output files' names will change accordingly. If you rename unbind.namd script to a different file, its design will change the output files' names as well.

We recommend trying at least these two simulations:

- Change the starting structure from s0 to a different structure, for example the antiparallel dimer s8. If you keep using the "drag" restraint, update the initial and final center of the restraint in a way that is compatible with the new starting structure.
- Apply additional restraints to the internal structure of the two peptides, implemented by force_helical_structure.colvars.in. The role of these additional restraints may be essential to handle properly certain starting structures, for example s4.

How do the free energies of dissociation compare between different starting structures?
What would be a good method to connect all these states (besides a multi-dimensional calculation)?

3. Ideas for further work

You can try one or more of the following approaches:

1) After trying out the unbinding simulation, you can set up a simulation to steer the binding process itself. How does the irreversible work for this process compare with that for unbinding?
2) Remove the "drag" restraint, and use a *different biasing potential*: this can be for example a restraint moving to a region previously not sampled, the metadynamics potential, or the ABF algorithm. Several inputs are ready for this purpose in unbind.namd.
   How does the sampling coverage compare with steered MD?

   Note: it is generally possible to combine algorithms, keeping in mind that applying two or more time-dependent biasing forces concurrently can lead to inconsistent results (unless you know how to account for the compounded effect of all biases). In general, the safest approach is to use only one bias that changes over time, while all others are constant restraints that will be included in any computed PMFs. Of course, restraints also change what is sampled, and they must be accounted for when interpreting the results.
3) Instead of enforcing a specific orientation and simulate helix-helix dissociation, you can use spinAngle variables (see example in the file packing_angles.colvars.in) to study the mutual orientation of both helices within the dimer. There should be significant free energy barriers between basins, each defined by a given peg-in-hole arrangement.

To explore all the options at your disposal, you can consult the reference manual for NAMD:

http://colvars.github.io/colvars-refman-namd/colvars-refman-namd.html

or download its PDF version:

http://colvars.github.io/pdf/colvars-refman-namd.pdf