

GROMACS Implementation of Metadynamics in Essential Coordinates

Contact:

Vojtěch Spiwok
Centre for Glycomics
Chemical Department, Slovak Academy of Sciences
Dúbravská cesta 9, 842 38 Bratislava, Slovakia
chemspiw@savba.sk, spiwokv@vscht.cz
web.vscht.cz/spiwokv, www.youtube.com/spiwokv

Thermodynamics as well as kinetics of chemical processes such as chemical reactions, crystal structure transitions or conformational changes in biopolymers can be described by a free energy surface. Free energy surface can be viewed as a function of free energy in the space of parameters which describe the progress of the studied chemical process. Accurate calculation of free energy surfaces of chemical processes is one of greatest challenges of computational chemistry. However, this task is extremely difficult due to poor sampling of configurational space by conventional molecular modelling methods.

Metadynamics¹⁻⁵ is recently introduced molecular simulation technique which enhances sampling of a configurational space. In metadynamics the system is simulated by a classical molecular dynamics simulation. During this simulation a history-dependent bias potential is being added. This bias potential disfavours previously visited regions of configurational space by “flooding” already visited free energy basins. When all free energy basins are flooded the resulting bias potential approximates the free energy surface of the studied system. Performance of this technique has been illustrated by numerous successful applications.

In order to apply metadynamics in modelling of conformational changes we introduced metadynamics in the space of essential coordinates⁶. Principle component analysis of molecular dynamics trajectories⁷ is often used to trace collective motions in biopolymers. Essential coordinates are projections of a conformation onto selected eigenvectors obtained by principle component analysis. Metadynamics in the space of essential coordinates can therefore model a conformational change in a biopolymer and provide its dimensionally-reduced free energy surface.

In the work which introduced metadynamics in essential coordinates we used the original formulation of metadynamics.¹ This was performed using the essential dynamics sampling utility of the standard version of GROMACS. Metadynamics was performed by a script written in Python language. Recently introduced formulation of metadynamics² referred to as Lagrangian turned out to be more efficient than the original formulation. In order to introduce this formulation into GROMACS package it was necessary to

introduced some changes to its code. Beside the Lagrangian formulation we have also implemented a similar formulation referred to as direct metadynamics⁴⁻⁵.

The first step of metadynamics is a selection of collective variables. In the metadynamics in essential coordinates these variables are projections of a conformation onto the selected eigenvectors, each representing a collective motion. Trajectory for essential dynamics analysis can be obtained by a classical molecular dynamics simulation. Then the studied biopolymer or its selected atoms are superimposed to remove translational and rotational motions. Then a covariance matrix is calculated, defined as:

$$C_{ij} = \langle (x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle)^T \rangle$$

where $\langle \rangle$ denotes averaging over trajectory. The covariance matrix is then diagonalized:

$$\Lambda = \mathbf{E} \mathbf{C} \mathbf{E}^{-1}$$

where \mathbf{E} is a matrix of eigenvectors and Λ is a diagonal matrix of eigenvalues. Eigenvectors describe collective motions whereas their eigenvalues describe their contribution to overall motions. Projection of a structure onto an i -th eigenvector is defined as:

$$s_j(t) = (\mathbf{x}(t) - \langle \mathbf{x} \rangle) \mathbf{e}_j^T$$

where \mathbf{x} are coordinates of the biopolymer superimposed onto reference structure.

In metadynamics a bias potential is defined as:

$$V_{bias}(\mathbf{z}, t) = \sum_i \prod_j w_i \exp\left(-\frac{(z_j - z_{ij}^{hills})^2}{2 \delta z_{ij}^2}\right)$$

where i is the index of a hill, j is the index of an essential coordinate, w is height, δz is width and z^{hills} is the position of a hill. Meaning of the vector \mathbf{z} is different for the Lagrangian and the direct metadynamics. In Lagrangian metadynamics the system is driven by the Lagrangian:

$$L = L_{MDS} + \frac{1}{2} \sum_j M_j \left(\frac{\partial \pi_j}{\partial \tau} \right)^2 + \frac{1}{2} \sum_j \lambda_j (p_j(\mathbf{x}) - \pi_j)^2 + V_{bias}(\boldsymbol{\pi}, t)$$

where L is the Lagrangian, L_{MDS} is the Lagrangian of classical molecular dynamics simulation, \mathbf{p} is projection of the system to the space of collective variables and $\boldsymbol{\pi}$ is the position of fictive particle with mass \mathbf{M} located in the

space of collective variables. Beside the dynamics of atoms in the Cartesian space there is also a dynamics of the fictive particle in the space of collective variables. The dynamics of atoms is driven by a standard molecular mechanics potential and by the restraint to the fictive particle. The dynamics of the fictive particle is driven by the restraint to the system and by the bias potential. In direct metadynamics the bias potential is coupled directly to the system as described by Lagrangian:

$$L = L_{MDS} + V_{bias}(\mathbf{p}(\mathbf{x}), t)$$

IMPLEMENTATION

Metadynamics in essential coordinates was implemented in the similar manner as essential dynamics sampling⁸⁻⁹ or flooding¹⁰ and many parts of the code of essential dynamics sampling and flooding were also used in metadynamics. Changes are in files `/src/tools/make_edi.c`, `/src/kernel/mdrun.c`, `/src/mdlib/edsam.c`, `/src/mdlib/force.c`, `/src/gmxlib/filenm.c`, `/src/mdlib/update.c`, `/include/edsam.h` and `/include/types/edsams.h`, most of them in the file `/src/mdlib/edsam.c`. By default, the bias potential can comprise up to 20,000 hills. This can be changed by modification of the parameter `NHILLS` in `/src/mdlib/edsam.c`. Metadynamics can be performed in the space of four essential coordinates. This number can be also changed, but this requires more complex changes in the code. Please, let me know if you like to run metadynamics in the space of five or more essential coordinates. Input file for metadynamics in essential coordinates is modified `edi` file as already used by flooding and essential dynamics sampling. This file can be generated by the modified tool `make_edi`.

No special attempts to introduce the multiple walker metadynamics¹¹ were made, but it was demonstrated that 10 jobs can run separately with a common bias potential (same as in tutorial). In future we want to improve it, e.g. by MPI version.

INSTALATION

The code can be installed in the same fashion as standard GROMACS code, i.e. by typing `./configure`, `make` and `make install` (see GROMACS manual or

web site for details). Despite the fact that we did our best to keep all other GROMACS functionalities unchanged, we strongly recommend users of this code to install it separately and to use the standard code for general molecular dynamics simulations.

PROGRAM OPTIONS

Here is the list of additional options of the programs `make_edi` and `mdrun`. They can be viewed also by running these programs with `-h` option.

`make_edi`:

<code>-mtdw</code>	real	1	height of a metadynamics hill
<code>-mtdlagrange</code>	int	0	direct metadynamics if 0, Lagrangian metadynamics if 1
<code>-mtdrestart</code>	int	0	$s(t=0) = S(r(t=0))$ if 0, S given explicitly if 1
<code>-mtdperstep</code>	int	1000	every <code>mtdperstep</code> a hill is added
<code>-mtds</code>	string	0.1,0.1,0.1,0.1	initial value(s) of essential coordinate(s)
<code>-mtdss</code>	string	0.1,0.1,0.1,0.1	width(s) of hill(s)
<code>-mtdlambdas</code>	string	100,100,100,100	value(s) of spring constant(s) in Lagrangian metadynamics
<code>-mtdmasses</code>	string	100,100,100,100	value(s) of mass(es) in Lagrangian metadynamics

`mdrun`:

<code>-mtd</code>	<code>mtd.mtd</code>	Output, Opt.	MTDEC output
-------------------	----------------------	--------------	--------------

TUTORIAL

First we have to obtain a trajectory for essential dynamics analysis. The file `mtdec_tutor.tar.gz` contains coordinate and topology of solvated alanine dipeptide. You need Amber ports¹² installed. The file can be unpacked by:

```
tar -xzf mtdec_tutor.tar.gz
```

The classical nanosecond molecular dynamics is performed by typing:

```
grompp -f mds -c before_mds -p acealanme -o mds
mdrun -s mds -o mds -g mds -e mds -c after_mds
```

This takes some 60 minutes on a single CPU (Xeon® 2.4 GHz). Resulting trajectory is then analysed by the program `g_covar`. We have good experience with using of five atoms to define essential coordinates, namely the methyl

and the carboxyl carbons of the acetyl, the C α atom of the alanine and the nitrogen and the carbon atoms of methylamide. These atoms must be first selected to the index file by typing:

```
make_ndx -f before_mds -o
```

The program asks for atoms. Type a1|a5|a9|a17|a19 and then q to quit the program. Essential dynamics analysis is performed by typing:

```
g_covar -s after_mds -f mds -n -o -v -av -l
```

The *edi* file is created by the modified version of *make_edi* by:

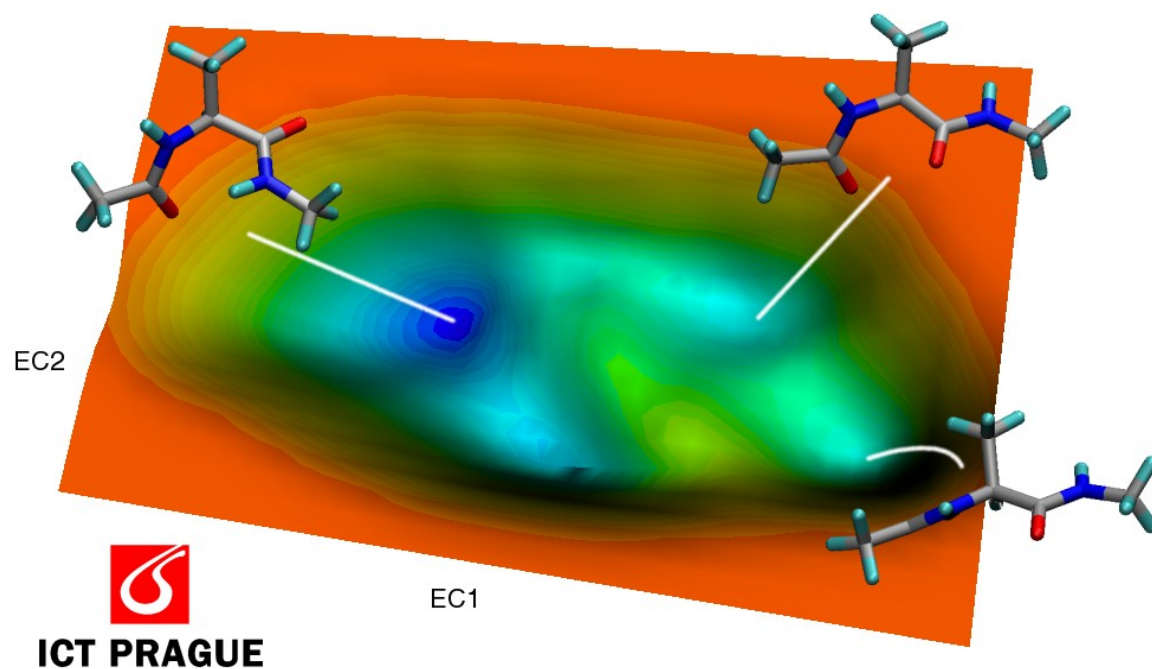
```
make_edi -s after_mds -f -n -o -mtdec 1,2 -mtdw 0.2 \  
-mtdperstep 1000 -mtdss 0.01,0.01
```

The resulting *sam.edi* file should contain lines with parameters of a direct metadynamics in essential coordinates (2 ns). It can be run by typing:

```
grompp -f mtd -c after_mds -p acealanme -o mtd  
mdrun -s mtd -o mtd -g mtd -e mtd -c after_mtd -ei sam -eo sam -mtd hills
```

Speed of metadynamics was 93 % of speed of molecular dynamics. Hills are written every 1,000 steps. The first column contains the walker number (currently set to 1, will be changed in future versions in more advanced multiple walker metadynamics). The second row contains the step number. The third column contains height of hills in kJ/mol. Other columns contain coordinates and widths of hills in nanometers. If two coordinates are used, then the first and the second columns contain coordinates and the third and the fourth contain widths.

In order to visualize the resulting free energy surface you can use some script to convert it to the form of free energy surface (we use Python scripts or Python with computationally difficult parts written in C implemented using SWIG). The resulting data can be viewed by some visualization tool such as Matplotlib, Mayavi, MapleTM etc. The resulting FES should look like one in Fig. 1 (visualized using Mayavi). It is possible, that your FES is flipped horizontally, vertically, or in both coordinates because positive and negative eigenvalues are equivalent in formation of a basis of collective motions. Video showing the trajectory and flooding can be found at <http://www.youtube.com/spiwoky>.

Figure 1. Conformational free energy surface of Ace-Ala-Nme.

REFERENCES

1. Laio A, Parrinello M (2002) *Proc Natl Acad Sci USA* **99**, 12562-12566.
2. Iannuzzi M, Laio A, Parrinello M (2003) *Phys Rev Lett* **90**, 238302.
3. Laio, Rodriguez-Foreta A, Gervasio FL, Ceccarelli M, Parrinello (2005) *J Phys Chem B* **109**, 6714-6721.
4. Bussi G, Laio A, Parrinello M (2006) *Phys Rev Lett* **96**, 090601.
5. Ensing B, de Vivo M, Liu Z, More P, Klein ML (2006) *Acc Chem Res* **39**, 73-81.
6. Spiwok V, Lipovová P, Králová B (2007) *J Phys Chem B* **111**, 3073-3076.
7. Amadei A, Linssen AB, Berendsen HJ (1993) *Proteins* **17**, 412-425.
8. Amadei A, Linssen AB, de Groot BL, van Aalten DM, Berendsen HJ (1996) *J Biomol Struct Dyn* **13**, 615-625.
9. de Groot BL, Amadei A, van Aalten DM, Berendsen HJ (1996) *J Biomol Struct Dyn* **13**, 741-751.
10. Lange OF, Schafer LV, Grubmuller H (2006) *J Comput Chem* **27**, 1693-1702.
11. Raiteri P, Laio A, Gervasio FL, Micheletti C, Parrinello M (2006) *J Phys Chem B* **110**, 3533-3539.
12. <http://chemistry.csulb.edu/ffamber/>