pyMDMix *The python package for your organic mixtures simulations*

Other tools

pyMDMix incorporates some utility commands to perform a variety of operations. All these commands are grouped under **mdmix tools** command. Printing help gives an overview of the functions implemented:

```
> mdmix tools -h
 ==========================================================
 ||              pyMDMix User Interface                  ||
 ==========================================================
 || Author: Daniel Alvarez-Garcia                        ||
 || Version : 0.1                                        ||
 ==========================================================
usage: mdmix tools [-h] {extendsim,projecttemplate,getvalues} ...
positional arguments:
 {extendsim,projecttemplate,getvalues}
                         Tool commands
    getvalues            Get energy values from grids at given coordinates.
    projecttemplate      Write template for project configuration
    extendsim            Extend simulation length for selected replicas
    diffgrids            Save new grid with the subtraction from two other
                         (grid1 - grid2)
    trim                 Use a PDB file as reference to crop/trim a grid and
                                          save just a part of the space (useful to reduce sizes
                                          and save disk space). WARNING: There might be a slight
                                          shift of coordinates in the trimmed grid.
    sumgrids             Save new grid with the addition of two other (grid1 +
                         grid2)
optional arguments:
   -h, --help            show this help message and exit
```

Write a project template

Use this command to save a local copy of a project configuration template which includes SYSTEM and MDSETTINGS sections to help your project set up. This command will save a file called `myproject.cfg` that the user can fill in:

```
> mdmix tools projecttemplate -f myproject.cfg
```

Extend simulations

Add nanoseconds to the selected replicas in order to extend the simulation. `COMMANDS.sh` file will be updated and needed input files generated. The command should be executed inside the project folder and is constructed like this:

```
> mdmix tools extendsim [selection] -n EXTRANANOS
```

where `[seletion]` follows same syntax for selecting the replicas to be modified as explained in the Analysis Guide. `EXTRANANOS` is the number of nanoseconds to add to the simulation length (e.g. if we had a simulation of 20ns and we want 50ns, we need to add 30ns). Example:

```
> mdmix tools extendsim all -n 50
```

This will add 50 ns to all replicas in the project. You can check the total length now considered printing the project information (mdmix info project).

You will need to create again queueing system input files for the new steps of the extended simulation as this is not done automatically.

Fetch energy values from a grid

For many analysis, it is useful to know the values that a grid takes in particular 3D coordinates. For this reason, this tool has been implemented to query a grid file with a set of coordinates defined by the user and return the energy values there contained. It is also possible to calculate the mean energy value over a sphere and not just a point. To do so, the point coordinates can be accompanied by a radius value. In this situation, the energy value returned, will be the average over all the points enclosed by the sphere. *Values equal to 999 will be automatically excluded from the average (999 value is used by the program to mask points in the grid where the density was equal to zero as a surrogate of excluded volume information).*

Two ways for defining the points to query have been implemented:

1. By means of a text file in which each line corresponds to a space separated list of x, y, z cordinates. Optionally, a 4th value can be added to each line corresponding to the radius defining a sphere.
2. By means of a PDB file in which all the coordinates will be used as points to be queried and b-factors column will be used to define radius and query the grid using spheres. If b-factor column is zero, only points will be queried.

Remember you can always get help on the command to know the options and flags to use (`mdmix tools getvalues -h`).

Query a grid using a coordinate text file

```
> mdmix tools getvalues -i INGRID -o OUTFILE -xyz XYZFILE [-r RADIUS]
```

Where `OUTFILE` can be any file name where results will be saved, `RADIUS` is an optional optional argument which will be explained later in the examples and `XYZFILE` is an ascii file written by the user which contains a set of x, y, z coordinates plus an optional radius per line (values separated by spaces). E.g.:

```
# Input coordinate file points.txt
-18.000    2.500    1.500
-12.000   14.500    0.500
```

Or with a radius for each coordinate:

```
# Input coordinate file spheres.txt
-18.000    2.500    1.500 1.5
-12.000   14.500    0.500 2.0
```

Examples:

```
> mdmix tools getvalues -i PROBE_AVG/ETA_ETA_CT_DG0.dx -o energiesCT.txt -xyz points.txt
```

Will write energy values for the two points declared in `points.txt` inside `energiesCT.txt` file. One energy value per line with same order as the input coordinates.

```
> mdmix tools getvalues -i PROBE_AVG/ETA_ETA_CT_DG0.dx -o energiesCT_spheres.txt -xyz spheres.txt
```

This will save mean energy values at the spheres defined in `spheres.txt`. Mean energy values will be saved in `energiesCT_spheres.txt` one per line matching the same order as in the input `spheres.txt`.

The optional -r RADIUS flag will override any behavriour defined in the cooridnates file. That is, even we have set a radius after the xyz coordinates, the value of the flag will override it. Therefore, we could call the following command to use the points.txt file as a sphere method by giving a common radius:

```
> mdmix tools getvalues -i PROBE_AVG/ETA_ETA_CT_DG0.dx -o energiesCT_spheres2.txt -xyz points.txt -r 2.0
```

The results will correspond to the spheres defined at centers in points.txt with a radius of 2.0 angstroms. In the same manner, if this was executed with spheres.txt as input, the 2.0 angstroms radii will be used instead of the radii values defined in the file.

Query a grid using a PDB file

```
> mdmix tools getvalues -i INGRID -o OUTFILE -pdb PDBFILE [-r RADIUS]
```

Where `OUTFILE` can be any file name where results will be saved, `RADIUS` is an optional optional argument which will be explained later in the examples and `PDBFILE` is a PDB format file where each atom position will act as a coordinate to query. The b-factor column in the PDB will define a radius to be used for querying a sphere instead of a single point. E.g.:

```
# File saved as points.pdb
ATOM 1 C MAC A 1 -3.000 3.500 -5.000 0.00 0.0
ATOM 2 C MAC A 1 -2.500 3.000 -5.500 0.00 0.0
ATOM 3 C MAC A 1 -2.500 3.000 -5.000 0.00 0.0
```

In this case, b-factor column is set to zeros and the query will return the point values. The PDB format must be respected, thus atom record should have correct format even the residue and atom names and numbers are not used.

```
# File saved as spheres.pdb
ATOM 1 C MAC A 1 -3.000 3.500 -5.000 0.00 1.5
ATOM 2 C MAC A 1 -2.500 3.000 -5.500 0.00 2.0
ATOM 3 C MAC A 1 -2.500 3.000 -5.000 0.00 2.0
```

In this second example the previous coordinates will now behave as spheres with radius 1.5, 2.0 and 2.0 angstroms as defined in the last column (b-factor column).

Examples:

```
> mdmix tools getvalues -i PROBE_AVG/ETA_ETA_CT_DG0.dx -o energiesCT.txt -pdb points.pdb
> mdmix tools getvalues -i PROBE_AVG/ETA_ETA_CT_DG0.dx -o energiesCT_spheres.txt -pdb spheres.txt
```

The first command will fetch point energies at each tomic position of the PDB file and the second will average the values enclosed by the spheres defined in the PDB with the radius of the b-factor column.

As in the previous case with coordinates file, this behavior can be overridden by specifying a particular radius from the command line:

```
> mdmix tools getvalues -i PROBE_AVG/ETA_ETA_CT_DG0.dx -o energiesCT_spheres2.txt -pdb points.txt -r 2.0
```

This will make the points in the pdb a center of a sphere with 2.0 radius and return an average of the values enclosed.

Grid difference

Calculate a difference grid between two input ones. Values above zero will be converted to zero before subtraction. The output will be saved to a new grid file that can be visualized. In this file, negative values will represent regions where the first grid is more favorable than the second and the opposite: positive values mean regions where the second grid is more favorable than the first one.

Usage:

```
> mdmix tools diffgrids -g1 [grid1 path] -g2 [grid2 path] -o [outfile]
```

Where the subtraction `grid1 - grid2` will be saved to file with name `outfile` in DX format unless the extension `.xplor` is given to the file name (in this case, XPLOR format will be used).

Grid addition

Sum two grids and save result to a new file with DX format (unles extension .xplor is used for the file name):

```
> mdmix tools sumgrids -g1 [grid1 path] -g2 [grid2 path] -o [outfile]
```

Grid trimming

Often we are interested in a region of the protein (e.g. the binding site) and handling huge grid files becomes tedious. With this function it is possible to trim a big grid and make it smaller. To tell the program what region of the space we are interested in give a PDB file with some atoms defining the region in space (it does not need to be a valid molecule, any set of coordinates will do it). The program will calculate the minimum and maximum coordinate to define the boundaries of the space to be extracted. A buffer distance in angstroms will be added either side (by default 8 Ansgtroms).

```
> mdmix tools trim -i [input grid path] -o [output grid filename] -pdb [reference pdb for trimming] [--buff 8.0]
```