pyMDMix

The python package for your organic mixtures simulations

MDMix method pyMDMix Documentation Blog

Analysis Guide

Content

Overview

IMPORTANT: for the analysis process to work, the simulation output must be moved to the project directory and filenames must match the expected ones (check COMMANDS.sh file inside the replica folder). It is not possible to analyze simulations run without the folder structure (at the moment).   When simulation is finished, we can start analyzing results. The process commonly involves the following steps:

1. Align the trajectory of different replicas to a common reference structure. Check if alignment was correctly done.
2. Compute density maps for each probe in the simulated solvent mixture.
3. Convert densities to energy maps (averaged over different replicas or for individual replicas).

Before entering into the analysis process, let me introduce you the selection algebra for selecting what replicas are going to be analyzed and other common options to the majority of analysis commands.

Replica selection for analysis commands

Most of pyMDMix analysis commands require the user to specify which replicas are going to be analyzed. The construction of such command has a shape similar to this one:

```
> mdmix analyze [command] [selection] [command-options]
```

Where `command` depends on the analysis stage (alignment, density calculation, energy conversion, etc.) and `command-options` are command specific options that the user can always check by getting help (`mdmix analyze [command] -h`). Common `command-options` include the number of processors to use for the analysis or the nanoseconds to analyze (if we are interested in a sub-selection of the trajectory). Selection methods:

- **`all`**: All replicas in the project will be processed.
- **`bysolvent -s [solvent list]`**: Giving a list of solvent box names, all replicas belonging to those solvents will be analyzed.
- **`byname -s [replica name list]`**: Give a list of replica names and only those replicas will be analyzed
- **`group -s [group name]`**: Give a name of a group and replicas belonging to that group will be analyzed. The group must be defined beforehand.

Examples:

We want to align trajectory of all replicas:

```
  > mdmix analyze align all
```

We want to calculate density maps for replicas with MAM solvent mixture:

```
   > mdmix analyze density bysolvent -s MAM
```

We want to convert to energies all density maps in replicas ETA_1 and ETA_2 specifically:

```
   > mdmix analyze energy byname -s ETA_1 ETA_2
```

We want to calculate density maps for a group we created before:

```
   > mdmix analyze density group -s ethanol_free
```

Step selection and number of processors to use

The simulation input files will be prepared to write one file for each nanosecond of simulation by default. It will be possible therefore to select a subset of nanoseconds to analyze by using -N option. If you change the default settings and produce trajectory files with larger number of nanoseconds, instead of nanosecond selection, it will be a step selection (as the program is only able to work with whole files and not portions of it). In any case, the command will be the same. To check what analysis commands include this possibility, call the command with -h flag to get help. For instance, we might be interested in aligning the trajectory only on the first four nanoseconds (or steps) for all replicas belonging to ETA solvent (i.e. because the rest are still running):

```
> mdmix analyze align bysolvent -s ETA -N1:4
```

Being the colon a range maker: all numbers from 1 to 4 in this case. Besides the step selection, some analysis commands allow parallel execution using multiple processors (e.g. density command). In this case, we should tell the program how many processors to use with -C option.

```
> mdmix analyze density bysolvent -s ETA -C8
```

This command will analyze all replicas belonging to ETA solvent, all known steps, using 8 processors.

Trajectory alignment

1. Theory

Trajectory alignment is performed by interfacing with ptraj(cpptraj) software from AmberTools. For each step, one script will be created inside `align/` subfolder in the replica directory. This script will perform three actions:

1. Align and image the trajectory over a reference structure (the reference pdb saved inside replica folder) using the the residue's backbone atoms selected with `ALIGNMASK` (defined when replica was created through the MD Settings or by giving a mask when executing the command) and save the new trajectory in align folder.
   The imaging commands are automatically calculated based on the number of chains for the solute (the protein or NA). For each chain, an imaging command will be issued.
   ATTENTION: The imaging process can take a long time depending on the system, so I highly recommend to carefully inspect and edit these commands if the automatic setup is not optimal for your system. E.g.: in many chained-proteins and applying restraints, there is no need for an imaging process for each chain.
2. Calculate an RMSD for the backbone and heavy atoms of the residues in the alignment mask and output results in replica align/ folder. These files can be automatically plotted using plotting commands (see last section in this page).
3. Calculate an average protein structure for each step and output a PDB file for each step in align folder. This is useful for checking if the protein underwent conformational changes.

Once trajectory is aligned, it is recommended to check the imaging run correctly by plotting rmsd files (see next section). It can happen that imaging is not correct for proteins with multiple chains. In this case, the scripts should be manually adapted to correct this bad behaviour (it is sometimes quite complex to automatically image multi-chained proteins). If trajectory is correctly aligned, it is possible to remove the original trajectory files. Please, keep output records for each step (will be needed to plot properties). As an example, the input script for ptraj can look like this:

```
# md1.ptraj script
trajin ../md/md1.nc
reference ../brd3_ETA_ETA_1_ref.pdb
center :1-129 mass origin
image :* origin center byres familiar
rms reference ":1-129@CA,C,N,O" out md2_bb_rmsd.out
rms reference nofit ":1-129 & !@H=" out md2_ha_rmsd.out
average prot_avg_1.pdb :1-129 pdb
trajout md1.nc netcdf
```

2. Usage

These scripts are automatically created and executed by calling align analysis command:

```
> mdmix analyze align [selection] [align-options]
```

Help on align options can be obtained:

```
> mdmix analyze align -h
 ===========================================================
 ||              pyMDMix User Interface                   ||
 ===========================================================
 || Author: Daniel Alvarez-Garcia                         ||
 || Version : 0.1                                         ||
 ===========================================================

usage: mdmix analyze align [-h] [-s SELECTION [SELECTION ...]]
                                  [-N NANOSELECT] [-C NCPUS]
                                  {all,bysolvent,byname,group}
positional arguments:
   {all,bysolvent,byname,group}
                          Perform selection of replicas based on solvent name,
                          replica names or groups. If 'all', do action on all
                          replicas.
optional arguments:
 -h, --help                     show this help message and exit
 -s SELECTION [SELECTION ...]
                                Selection list. If selecting 'bysolvent', list of
                                solvent names is expected. If 'byname', list of
                                replica names. If 'group', group name. Skip if 'all'
                                is selected.
 -N NANOSELECT                  List production steps to consider for alignment using
                                a colon separated range. Ex: 1:20 - first to 20th step.
 -C NCPUS                       Number of cpus to use for the action. If option not
                                given, will use 1 CPU serial mode.
--mask ALIGNMASK                Modify alignment mask defined when creating the
                                replicas. By default the macromolecule will be
                                automatically identified. Give a list with comma
                                separated residue numbers or hyphen separated range.
                                E.g. 10-100,120-240.
 --ref REF                      Path to reference PDB file. By default, pyMDmix
                                generates one automatic reference pdb file which can
                                be found inside each replica folder. This option will
                                override it.
--only-write                    Only write ptraj input scripts BUT don't execute them.
                                Useful when manual editing is needed. (default: False)
--only-exe                      Only execute existing ptra scripts, do not overwrite
                                them. If scripts don't exist, this function will fail.
                                (Default: False)
```

3. Examples

The following example will align all replicas’ trajectories for solvent WAT using 4 cpus:

```
> mdmix analyze align bysolvent -s WAT -C4
```

For each cpu given, an independent ptraj execution will be called with one of the input scripts generated. It is not possible to run more parallel ptrajs than scripts created (i.e. if we have only 1 replica with 4 nanoseconds and 4 trajectory files, only 4 parallel executions can be run).

```
> mdmix analyze align all -N1:10
```

In this example, only trajectory files 1 to 10 for all replicas in the project will be aligned. Useful when the simulation is still running: we can start aligning already finished steps. For all previous examples, the trajectory will be aligned over the residues defined when the replica was created using MD Settings ALIGNMASK option. If it was not defined, the program will automatically identify the macromolecule residues and align over all of them using only the backbone atoms. This default behaviour can be now altered by giving another mask when calling the command. For instance, we want to align over backbone atoms in residues 5 to 190 only for replica named ETA_1 (notice the change of colon : to hyphen -, this is an amber mask format):

```
> mdmix analyze align byname -s ETA_1 --mask 5-190
```

Bear in mind that in this case, RMSD output data will be only for the aligned region.

```
> mdmix analyze align all --only-write
```

This command will write the input ptraj scripts for all replicas in the project **but will not execute them**. This is useful to finely tune the scripts and execute them later.

```
> mdmix analyze align all --only-exe -C8
```

This command is the follow-up of the previous one. Will execute the ptraj scripts found in the replica alignment folder (for all replicas) and execute the processes (using 8 CPUs). If the ptraj input files are not found, the execution will fail.

Plotting commands

Some automatic plotting functions have been implemented to check if the simulation run correctly and the alignment process was correct.

Amber plots

We can plot different attributes from the molecular dynamics as follows (**Attention**: this will only work if the simulation was run using AMBER. Moreover simulation output files should match expected names and be placed in *md* directory in each replica folder):

```
> mdmix plot ambermd [selection] -o [outname]
```

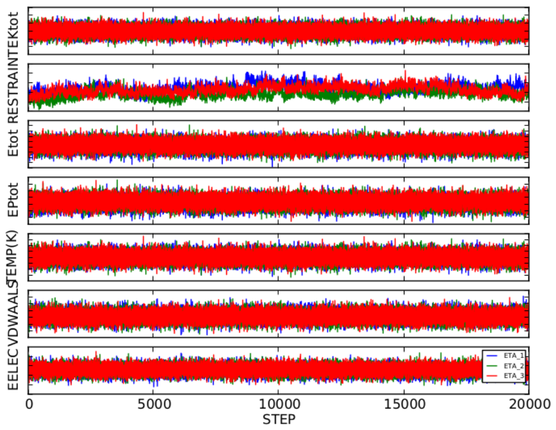
This will plot a predefined set of properties from the selected replicas. Here selection of replicas follows the standard selection syntax explained at the beginning of this analysis guide. The plot will be saved in a file called `[outname]`. One can choose in which file format the image will be saved simply by giving one of these extensions to the output file name: jpeg, png, pdf, ps or eps. By default, these amber fields will be plotted for each replica in the selection: `Etot, EPtot, EKtot, TEMP(K), RESTRAINT, VDWAALS, EELEC`. In this example these properties will be plotted for all replicas with ETA solvent and saved in a PDF file with name `mdplots.pdf`:

```
> mdmix plot ambermd bysolvent -s ETA -o mdplots.pdf
```

If the simulation is not finished yet but you have partially aligned some part of thetrajectory or even you want to check the energetics of already finished steps, you can add a step selection flag to plot only selected steps. With the following commands, only steps 1 to 10th will be plotted:

```
> mdmix plot ambermd bysolvent -s ETA -o mdplots.pdf -N1:10
```

For a correct simulation we should see a stable evolution like in the following plot:

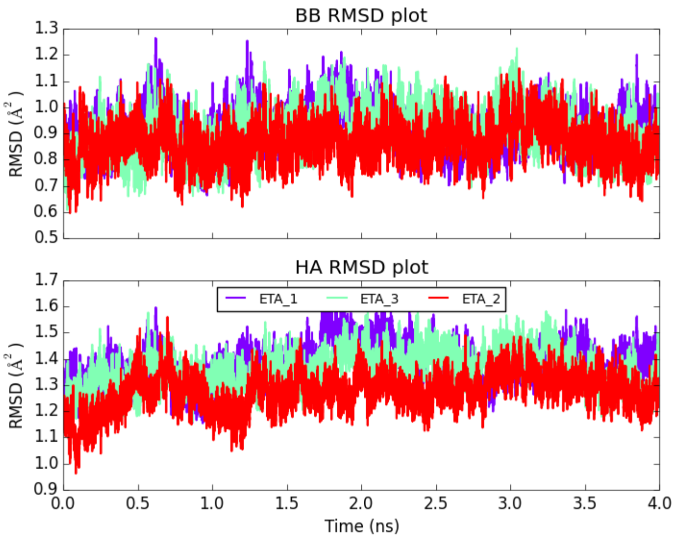


RMSD plot

Once trajectory is aligned, we can also plot an RMSD for the protein backbone and heavy atoms automatically. A file with the data plotted can be found inside each replica folder under `align/` subdirectory. For instance, the following command will plot RMSD values along all the trajectory of replicas belonging to ETA solvent and save the image as a PNG file with name `eta_rmsd.png`. As in the previous plotting example, one can choose the format of the output file simply by changing the output filename extension (possible extensions: `jpeg, png, pdf, ps, eps`).

```
> mdmix plot rmsd bysolvent -s ETA -o eta_rmsd.png
```

The RMSD plot should be stable (meaning no big jumps in the RMSD are observed which could mean a missalignment or bad imaging of the trajectory). A possible output plot could look like this:



Density calculation

1. Theory

The aligned trajectory (yes, it should be previously aligned for this command to work) will be analyzed to obtain a density map for each probe in the solvent (probes are defined for each solvent when the solvent database is created; they represent one or multiple atoms inside the same residue in the solvent). The process will build a static grid over the protein and count how many times each atom in the probe falls into each grid voxel along all the trajectory. The resulting grids will be saved inside the replica subfolder dgrids and can be visualized with common molecular visualization programs (PyMOL, VMD, Chimera…). Use the system reference PDB file found at project folder or replica folder for visualizing the protein structure in grid coordinates. Grid files will be named according to the replica and probe name: REPLICA_PROBE.dx (e.g. ETA_1_ETA_OH.dx).

2. Usage

```
> mdmix analyze density [selection] [density-options]
```

Selection method are described above and density-options are described when calling command help:

```
> mdmix analyze density -h
 ==========================================================
 ||                  pyMDMix User Interface              ||
 ==========================================================
 || Author: Daniel Alvarez-Garcia                        ||
 || Version : 0.1                                        ||
 ==========================================================

usage: mdmix analyze density [-h] [-s SELECTION [SELECTION ...]]
                                   [-N NANOSELECT] [--step STEP] [-C NCPUS]
                                   [--probes PROBELIST [PROBELIST ...]]
                                   [-opref OUTPREFIX] [--onlycom] [--com]
                                   {all,bysolvent,byname,group}
positional arguments:
    {all,bysolvent,byname,group}
                          Perform selection of replicas based on solvent name,
                          replica names or groups. If 'all', do action on all
                          replicas.
optional arguments:
 -h, --help               show this help message and exit
 -s SELECTION [SELECTION ...]
                          Selection list. If selecting 'bysolvent', list of
                          solvent names is expected. If 'byname', list of
                          replica names. If 'group', group name. Skip if 'all'
                          is selected.
 -N NANOSELECT            For action commands, list production steps to consider
                          for the analysis using a colon separated range. Ex:
                          1:20 - first to 20th step.
 --step STEP              Take snapshots every STEP number. Default:1.
 -C NCPUS                 Number of cpus to use for the action. If option not
                          given, will use 1 CPU serial mode.
 --probes PROBELIST [PROBELIST ...], -P PROBELIST [PROBELIST ...]
                          Selection of probenames to calculate density for. If
                          not given, all probes for the solvent will be
                          selected.
 -opref OUTPREFIX         Prefix for grids saved inside density folder for each
                          replica. If this is given, automatic energy conversion
                          will not work until you restablish expected names or
                          explicitely give the prefix in energy command.
                          Default: False
 --onlycom                Density calculations ONLY for center of masses of each
                          co-solvent. Probe list is ignored. Default: False
 --com                    Include center of mass to list of probes probes in
                          density calculation. Default: False
```

3. Examples:

Obtain density grids for all replicas of solvent ETA using 8 cpus. Grids will be saved with default names in dgrids folder inside each replica:

```
> mdmix analyze density bysolvent -s ETA -C 8
```

Calculate density grids for ONLY `ETA_OH` probe in all replicas of solvent ETA using 8 cpus (you can know what probes are available for a solvent by calling mdmix info solvents, check also solvents page):

```
> mdmix analyze density bysolvent -s ETA -C 8 -P ETA_OH
```

Besides mixture probes (defined when each solvent mixture was added to the program) pyMDMix will automatically identify the different molecules in the mixture and calculate a density grid for their center of mass position (COM) when demanded. E.g. ETA mixture contains two molecules or ‘residues’ with names ETA and WAT; if requested, new virtual probes `ETA_COM` and `WAT_COM` will be calculated and saved along the other probes. Execute with `-com` option to include these calculations to the standard probes:

```
> mdmix analyze density bysolvent -s ETA --com
```

One may be interested in calculating the density grids for the center of mass of each residue in the mixture. This is done with the following command:

```
> mdmix analyze density bysolvent -s ETA --onlycom
```

Advanced usage: Optionally, a subsample of the trajectory can be selected with `-N` option. In this example, the program will calculate only density grid for nanoseconds 10 to 20 (with default md settings in which 1 trajectory file means 1 nanosecond). In this case, to not mix results with the long trajectory analysis results, we prepend a prefix to the output file names (`partial_dens_`). Grids will still be saved inside each replica `dgrids/` directory but with this prefix.

```
> mdmix analyze density byname -s ETA_1 -N 10:20 -opref partial_dens_
```

Energy conversion

1. Theory

Considering the solvent configurational space over the protein has been correctly sampled, one can apply the Boltzmann relationship to convert the observed density distribution into free energy. By comparing this observed distribution (`Ni`) with an expected one (`N0`, which correspond to the uniform distribution expected in the bulk), a free energy of binding can be calculated. This expression is used to convert from densities to energies:

$$\Delta G_{bind} = -RTln\frac{N_i}{N_o} + \left(-RTln\frac{V_{sim}}{V_{1M}}\right)^*$$

R is the gas constant and **T** the temperature of the simulation. Second part in brackets is an analytical term introduced to correct for the bias originated by the higher concentration of our solvent mixtures with respect standard state conditions (volume of simulation is compared with the volume expected at 1M concentration). This second term is only applied for non-water probes.

Grid file names saved will identify the probe and whether this correction has been applied or not: energy grids without the correction will be named as PROBE_DG.dx, energy grids with the correction will have a DG0 instead of DG (PROBE_DG0.dx). E.g.: ETA_CT_DG0.dx for CT probe or ETA_WAT_DG.dx for water in ETA mixture.

2. Usage

As with previous analysis commands, the syntax follows a replica selection part and the command specific options:

```
> mdmix analyze energy [selection] [energy-options]
```

Which calling help gives:

```
> mdmix analyze energy --help
 ==========================================================
 ||                 pyMDMix User Interface               ||
 ==========================================================
 || Author: Daniel Alvarez-Garcia                        ||
 || Version : 0.1                                        ||
 ==========================================================


usage: mdmix analyze energy [-h] [-s SELECTION [SELECTION ...]]
                                  [-nsnaps NSNAPS]
                                  [--probes PROBELIST [PROBELIST ...]] [-noavg]
                                  [-ipref INPREFIX] [-opref OUTPREFIX] [-nodg0]
                                  {all,bysolvent,byname,group}
positional arguments:
       {all,bysolvent,byname,group}
                        Perform selection of replicas based on solvent name,
                       replica names or groups. If 'all', do action on all
                       replicas.
optional arguments:
 -h, --help            show this help message and exit
 -s SELECTION [SELECTION ...]
                       Selection list. If selecting 'bysolvent', list of
                       solvent names is expected. If 'byname', list of
                       replica names. If 'group', group name. Skip if 'all'
                       is selected.
 -nsnaps NSNAPS        If given, use this number of snapshots for calculating
                       the expected number instead of the total number.
                       Useful when a subset of the trajectory is analyzed.
 --probes PROBELIST [PROBELIST ...], -P PROBELIST [PROBELIST ...]
                       Selection of probenames to convert. If not given, all
                       probes for the solvent will be converted.
 -noavg                By default, densities for the diferent replicas will
                       be merged before energy conversion and a single
                       replica-averaged energy map for each probe will be
                       saved at project folder. To save separately each
                       replica it's own energy grids, use -noavg. Energy
                       grids will then be saved inside each replica folder
                       independently.
 -ipref INPREFIX       If density grids were saved with a specific prefix,
                       give it here so the program knows what density grids
                       to take. Default: no prefix (predefined names).
 -opref OUTPREFIX      Prefix for output average grids. Default: no prefix
                       (predefined names).
 -nodg0                Disable standard state correction. Ignore
                       concentration issues
```

3. Examples

In practical terms, the user will convert the already calculated density grids to energies issuing this command:

```
> mdmix analyze energy byname -s ETA_1
```

All density grids found inside `ETA_1` replica directory (usually one for each probe in the solvent mixture used) will be converted to energies. The resulting grids will be saved inside `egrids/` subdirectory of the replica folder. If more than one replica is selected, the density grids will be added up before conversion (only if they belong to the same probe). For instance:

```
> mdmix analyze energy bysolvent -s ETA
```

will sum up all density grids for all replicas run with ETA solvent mixture and then convert the resulting grid to energy. Therefore, only one grid per probe will be saved in disk. In this case, the resulting grids, as they are average of different replicas, will be saved in the main project folder under `PROBE_AVG/` subdirectory.

It is possible to independently convert each replica in the selection instead of summing densities before conversion. This is done giving `-noavg` flag:

```
> mdmix analyze energy bysolvent -s ETA -noavg
```

Here, as in the first example, all replicas with solvent ETA will be independently converted and the resulting grid files will be saved in each replica’s `egrid/` subdirectory.

If you wish, the correction by volume (second term in the equation before) can be disabled to obtain the raw energies by giving `-nodg0` flag when calling any of these commands.

Advanced usage: When the density grids were calculated over a subselection of the trajectory, we must tell the program what are the non-standard names we used and more importantly, what is the number of trajectory snapshots taken into account when building the density grid. Following example in previous section, trajectory files from 10 to 20 were used to calculate densities. If defaults were not modified, it means 1000 snapshots per trajectory file, making up 10000 snapshots for the subset. Again, we might be interested in differentialy save these grids with a prefix:

```
> mdmix analyze energy bysolvent -s ETA -nsnaps 10000 -ipref partial_dens_ -opref partial_dens_
```

Hot spots identification

1. Theory

We define a Hot Spot as a high affinity interaction spot over the protein surface. In practice, pyMDMix identifies hot spots by applying a certain **cutoff** to the energy grids obtained in previous steps. All points in the grid below certain cutoff will become a hot spot. It is a way of discretize the huge information contained in a grid file. This **cutoff** value is automatically established for each grid by taking the lower 0.02 percentile of the points. That is, points are sorted from lower to higher energy and the 0.02% of lower energy points are taken. From them, the higher energy is set as the cutoff value. This default percentile value can be modified when calling the command. Optionally a hard cutoff value can be selected replacing this automatic adaptive cutoff. Each voxel with a value below or equal to the cutoff will be saved as a Cartesian coordinate with its associated energy. All set of points are then grouped to identify the actual hot spots using a hierarchical clustering method with a cut distance of 1.5 Angstroms (all points within 1.5 A will join the same cluster). The total energy for the identified hot spot is calculated by averaging the associated probabilities and converting back to energies following this formula:

$$\Delta G_{hotspot} = -RTln \left\langle e^{\frac{\Delta G_{point}}{-RT}} \right\rangle_{points}$$

The average hot spot value is then assigned to the lower energy coordinate within the cluster (considered from now on the center of the hot spot). When more than one grid is given as input, the process above is fulfilled for each of them. After identifying all hot spots centers, hot spots across different grids which lay below 1.5 Angstroms of distance of each other are clustered together and only the minimum energy one is kept. This way we can obtain a combined view of different probes inside a single file. **CAUTION**: Be aware that direct comparison of energies across probes of different solvents might be not correct enough as bigger probes will tend to have lower energies. This way, probes also favorable in a given position will always be masked out by the bigger solvent molecules. We always recommend visualizing the energy grids with the hot spots identified and do not directly discard a probe because there's another one with lower energy. Both can be correct.

2. Usage

This command differs from the previous by the fact that we will work with files and not replicas anymore.

```
> mdmix analyze hotspots create -i INGRID -o OUTPREFIX [-p PERCENTILE] [-H HARDCUTOFF] [--allpoints]
```

All flags in square brackets are optional. The grid file given as input will be discretized and saved as a PDB file with file name `OUTPREFIX.pdb`. Moreover, a second file will be saved for internal use with file name `OUTPREFIX.hset`. This second file is needed by calculating residence plots in next section. Complete description of the options is again obtained by calling the command help:

```
> mdmix analyze hotspots create --help
 ==========================================================
 ||               pyMDMix User Interface                 ||
 ==========================================================
 || Author: Daniel Alvarez-Garcia                        ||
 || Version : 0.1                                        ||
 ==========================================================

usage: mdmix analyze hotspots create [-h] -i INGRIDS [INGRIDS ...] -o
                                     OUTPREFIX [--allpoints] [-p PERCENTILE]
                                     [--hardcutoff HARDCUTOFF]
optional arguments:
 -h, --help            show this help message and exit
 -i INGRIDS [INGRIDS ...]
                       List of grid files to use for hotspots creation. If
                       multiple probes are given, a hot spot set with the
                       minimum energy probes is obtained.
 -o OUTPREFIX          Output prefix. A PDB with the hotspots and a pickle
                       file will be saved.
 --allpoints           Write all points belonging to the hotspots in the
                       output PDB instead of only the minimums with mean
                       energy
 -p PERCENTILE         Percentile of points to use for establishing the
                       cutoff value. Default: 0.02
```

```
  --hardcutoff HARDCUTOFF, -H HARDCUTOFF
                        Stablish a hard energy cutoff instead of the adaptive one.
```

3. Examples

We simulated a protein with 3 replicas of ETA solvent mixture. Trajectory was aligned, density grids calculated and an energy average for each probe in ETA solvent was saved in the project folder's `PROBE_AVG` subdirectory. With this command we would discretize ETA_CT probe to identify it's lower energy hotspots using all default parameters (executed from project folder):

```
> mdmix analyze hotspots create -i PROBE_AVG/ETA_CT_DG0.dx -o ETA_CT_hotspots
```

This will save `ETA_CT_hotspots.pdb` and `ETA_CT_hotspots.hset` in current folder. You can visualize the PDB with any molecular visualization software. The mean energy value of the hotspot is saved in the b-factor column. Open also the system reference structure for correctly interpreting the results. If `-allpoints` flag is given, the PDB will contain all the points which conform the different clusters identified (each residue number in the pdb identifies a different cluster or hot spot). In this case, the energy value of the b-factor column corresponds to the individual point value and not the average. With the following command, hotspots from different probes will be combined into a single file. Probes will compete and only the lower energy one will be kept.

```
> mdmix analyze hotspots create -i PROBE_AVG/ETA_CT_DG0.dx PROBE_AVG/ETA_OH_DG0.dx PROBE_AVG/ETA_WAT_DG.
```

Residence plots

1. Theory

One way to assess the completeness of sampling at a particular spot over the protein (and here we will be mostly interested in hotspots), is to study the solvent exchange. Even in hotspots with a high affinity for the ligand, we should still see some solvent exchange. If we do not see it, this could mean that the simulation was not run long enough (or if restraints were applied, the exchange pathway was artificially modified) and the energy values one could obtain for the grids could not be converged (see Alvarez-Garcia D. and Barril X. JCTC. 2014 for more info on this topic). This issue gains relevance when we are planning to use the quantitative information contained in the energy maps. If making a qualitative use, this sampling problem could be less relevant. To study the solvent exchange, the program evaluates what molecules are near a particular set of coordinates along a trajectory (that is, what molecules visit the place?). This set of coordinates is defined as a **hotspot,** using information obtained in previous steps or as a sphere (where the user defines a center and a tolerance). A second use of this tool is for studying residence times in a binding process when a complete sampling is granted. Residence times will allow the user to estimate several thermodynamic and kinetic parameters. Beware trajectories must be previously aligned.

2. Usage

As any other analysis command, we will have to select what replicas will be studied and then give some command specific options. Many options are shared amongst other analysis commands like number of cpus, nanoseconds to analyze, etc. To check specific options and descriptions, call the command help:

```
> mdmix analyze residence --help
usage: mdmix analyze residence [-h] [-s SELECTION [SELECTION ...]]
                               [-N NANOSELECT] [--step STEP] [-C NCPUS]
                               [--hpfile HPFILE] [--hpid HPID]
                               [--center CENTER [CENTER ...]]
                               [--tol TOLERANCE]
                               {all,bysolvent,byname,group}
positional arguments:
            {all,bysolvent,byname,group}
            Perform selection of replicas based on solvent name,
            replica names or groups. If 'all', do action on all
            replicas.
optional arguments:
  -h, --help                   show this help message and exit
  -s SELECTION [SELECTION ...]
                               Selection list. If selecting 'bysolvent', list of
                               solvent names is expected. If 'byname', list of
                               replica names. If 'group', group name. Skip if 'all'
                               is selected.
  -N NANOSELECT                For action commands, list production steps to consider
```

```
                                  for the analysis using a colon separated range. Ex:
                                  1:20 - first to 20th step.
  --step STEP                     Take snapshots every STEP number. Default:1.
  -C NCPUS                        Number of cpus to use for the action. If option not
                                  given, will use 1 CPU serial mode.
  --hpfile HPFILE, -hf HPFILE
                                  HotspotSet pickled file from 'analyze hotspots create'
                                  action.
  --hpid HPID, -id HPID
                                  Hotspot ID in HPFILE to define region to study.
  --center CENTER [CENTER ...], -ce CENTER [CENTER ...]
                                  Sphere center to define region to study. Give a space
                                  separated 3 float list.
  --tol TOLERANCE, -t TOLERANCE
                                  Tolerance in angstroms around sphere center or hotspot
                                  coordinates to consider as occupied space.
```

There are two ways for defining the region to study using this command:

1. **Defining a center and radius**. All molecules visiting the sphere will be identified along the trajectory.
2. **Giving a hotspot set and a hot spot ID**. When creating hotspots from grid files in previous section, two files were saved: a PDB file and a file with same name and extension .hset. This second file is the hotspot set needed in this study. The hot spot ID is the residue number in the pair PDB file. Opening the PDB in any molecular visualization program allows the user to choose what hotspot he/she is interested in by visual instpection. The residue number corresponding to the hotspot will be the hotspot ID in this command.

All these different usage will be clarified with some examples later. The output of the command is a text file and a png image with a plot. These two files will be saved inside each replica directory. The text file contains the residue identifiers for each molecule visiting the region in study for each snapshot analyzed. At the top of the file, you will find a match between residue identifiers used and residue names.

3. Examples

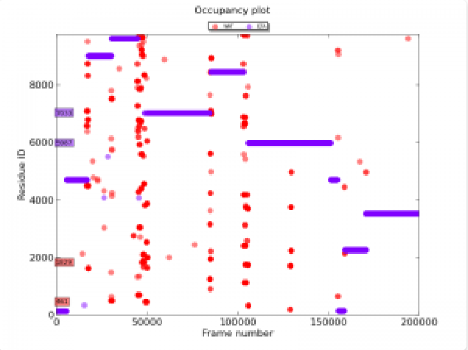
Study the exchange at a particular protein surface spot defined by a sphere (center at 16.0 17.0 32.0 and radius 1.5 angstroms). All replicas in the project will be studied at the same location. Will use 3 cpus for the analysis.

```
> mdmix analyze residence all -ce 16.0 17.0 32.0 -t 1.5 -C 3
```

In next example we will use a hot spot set and a identifier to define the region to study. In this case, we should be aware that all the points you see in the PDB for the same residue will act as spheres to identify molecules passing by. With this I mean that the tolerance in this case should be reduced to avoid considering a big region in the space (default 0.5 Angstroms will be OK in general). Here all replicas using solvent mixture ETA will be analyzed. The region to study is defined in HOTSPOTS.hset file which should be previously created (see previous section). Inside the hotspot set, we are interested in the hotspot 1 (which is the residue 1 in HOTSPOTS.pdb). Will use 4 cpus for the analysis.

```
> mdmix analyze residence bysolvent -s ETA -hpf HOTSPOTS.hset -hpid 1 -C 4
```

The result will be two files for each replica saved inside each replica folder independently. A tutorial with more examples and practical use will come soon. The resulting plots will look something like this:



Residence plot example