pyMDMix

The python package for your organic mixtures simulations

MDMix method pyMDMix Documentation Blog

MD settings

Introduction

MD settings refer to all parameters the program needs to set up a Replica (being a Replica an independent MD set up containing input files – topology and coordinates of the solvated system – and md configuration files – for minimization, equilibration and production steps -. pyMDMix needs a MD Settings Configuration File to create Replicas. This file will contain all parameters related to the simulation set up. Including:

1. **Solvent**: Add a truncated octahedron box with the solvent mixture selected to the system to simulate.
2. **Simulation parameters**: temperature, length in nanoseconds, trajectory writing frequency, MD simulation program (currently supported simulation programs are AMBER or NAMD.), steps per job, etc…
3. **Simulation restraints**: It is possible to automatically set positional restraints over the protein atoms. The residues to restrain are selected automatically or by using masks. Two options are predefined to select atoms in each residue that will be restrained: heavy atoms (`HA` option – all non-hydrogen atoms) or backbone atoms (`BB` option). A force value will be needed when restraining positions.
4. **Residue masks**: Specify with residues will be selected for applying positional restraints to (`RESTRMASK`) or residues over which the trajectory will be aligned at analysis stage (`ALIGNMASK`). If none of these options is specified at replica creation time, the program will try to detect the protein residues automatically.

The full list of options that can be customized is quite large and very often many parameters are rarely modified. For this reason, most of the options are pre-defined with a default value in pyMDMix. The user will not have to worry about many settings. In fact, the simplest and minimmum MD Settings Configuration File (MSC file) that can be written includes only the solvent to use:

```
[MDSETTINGS]
SOLVENT = ETA
```

With this input, a Replica for the selected project system will be created by solvating the system with an ethanol/water mixture called ETA (name should exist in Solvents Database). All MD parameters will take default values (e.g. 20 ns, 300K, without restraints, etc…). Each extra option given in this configuration file will override the defaults and create a unique set of settings for this replica (see examples below).

Default values can be found and modified inside two different files which are read in a hierarchical manner at replica creation time (find a copy of a template file at the bottom of this page which contains a simple description of each parameter).

1. **Package default settings (`$PYTHON_INSTALL_DIR`/pyMDMix/data/defaults/md-settings.cfg):** These default parameters are valid for all users in the platform. Modifications at this level will alter the defaults for all the program installation.
2. **User default settings (`$HOME/.mdmix/md-settings.cfg`)** : A copy of the package default settings file will be saved inside the user's home directory the first time pyMDMix is executed. Uncomment and modify any line you wish to alter. The options here defined will override the package defaults and will be valid only for current user projects.

Any option found in these two files can be included in the MSC file at replica creation time. Any option defined in MSC file has the highest priority and will override any user or package defaults. For instance, `md-settings.cfg` includes `trajfrequency` option (more precisely, found with INI syntax: `int-trajfrequency`). We could increase this value for the replica with ethanol we want to create with this MSC file:

```
[MDSETTINGS]
solvent = ETA
trajfrequency = 1000
```

MD Settings Configuration File (MSC file) for single replica definition

At this point, we can think of this file as the settings definition for creating a single Replica (see next how this same file can be used to define multiple replicas at once). The file which should be written by the user and adapted to the simulation requirements should is formed of a single section named MDSETTINGS followed by a list of option=value pairs.

Example 1: Simple replica setup

We want to simulate the system using ethanol mixture saved in solvent database with name ETA for 50 nanoseconds at 298K temperature.

```
[MDSETTINGS]
solvent = ETA
nanos = 50
temp = 298
```

Example 2: Replica setup with restraints

Same setup as before but now we want to apply positional restraints over all heavy atoms of the protein with a force of 0.5 kcal/mol·A2 to limit protein motion.

```
[MDSETTINGS]
solvent = ETA
nanos = 50
temp = 298
restr = HA
force = 0.5
```

Notice that we did not define what residues to restrain. The program will identify all protein residues automatically (at this point, it is important that non standard residues that you wish to be restrained are identified in the System Configuration file).

```
[MDSETTINGS]
solvent = ETA
nanos = 50
temp = 298
restr = HA
force = 0.5
restrmask = 3-150,155-290
```

In this last example we want to apply restraints to particular residues.

Example 3: Replica setup with longer simulation jobs

By default, the production stage is planned to include 1ns for each job run. That is, each nanosecond of simulation will be run independently and will output one trajectory file. If we want to include more nanoseconds per job run, we should modify **nvt_prod_steps** (nstlim in AMBER configuration files or numsteps in NAMD configuration). The default timestep is 2fs and default nvt_prod_steps is 500.000 (500.000 x 2 = 1.000.000 fs = 1 ns). For running 5ns per job run, prod_steps should be set to 2.500.000. At the same time, we want to ouput trajectory frames every 4ps (= every 2000 steps).

```
[MDSETTINGS]
solvent = ETA
nanos = 50
temp = 298
restr = HA
force = 0.5
restrmask = 3-150,155-290
prod_steps = 2500000
trajfrequency = 2000
```

MD Settings Configuration File (MSC file) for multiple replicas definition

It is possible to define multiple settings inside the same MSC file in two different ways:

1. Denining multiple MDSETTINGS entries
2. Using multivalue special options

1. Defininig multiple MDESTTINGS sections

We can create one replica with ethanol mixture and one with water using a single MSC file. E.g. we want the ethanol replica to run for 50ns with restraints on the heavy atoms but we one a replica with water without restraints and run it for 10ns only. The MSC file with multiple entries could be like this:

```
[MDSETTINGS1]
solvent = ETA
nanos = 50
restr = HA
force = 0.5

[MDSETTINGS2]
solvent = WAT
nanos = 10
```

Being MDSETTINGS1 and MDSETTINGS2 two arbitrary names that should be differential. Any identifier after MDSETTINGS word will make it (e.g. we could have called these sections MDSETTINGS_ETA and MSETTINGS_WAT). Remember all options not given will take default values (both will run at 300K temperature, water will have no restraints and restrains on the ethanol simulation will be applied automatically to the residues identified as part of the protein).

This way, we could add as many entries as replicas we want to create.

2. Definining multiple replicas with multivalue options in a single MDESTTINGS section

The previous example shows how adding multiple MDSETTINGS sections provides an intuitive way of creating multiple replicas with a single MSC file. If we want to add many replicas, this way is unpractical. That is why, some options have been modified to accept multiple values in the same entry and NREPL option has been added to modify the number of replicas with the same setup to be created:

- **SOLVENT(S):** (either with trailing S or without it). It is possible to define a list of comma-separated solvent identifiers and one replica will be created for each of them. E.g.: SOLVENTS = ETA, WAT
- **NREPL:** By default, when this option is not given, one replica will be created for each solvent in the previous option. We can increase this number to create more than one replica for each of them (e.g. NREPL = 3 will create 3 replicas for ETA and 3 more for WAT solvents).

For instance, if we want to run 3 replicas of ETA mixture all sharing the same MD setup, we could use the following MSC file:

```
[MDSETTINGS1]
solvent = ETA
nrepl = 3
nanos = 50
restr = HA
force = 0.5
```

Advanced differential set ups

Imagin we want to declare 3 replicas for ethanol and 1 for water. For this situation, NREPL has been designed to accept multiple assignments in the same option using a complex syntax. Let me show you one example:

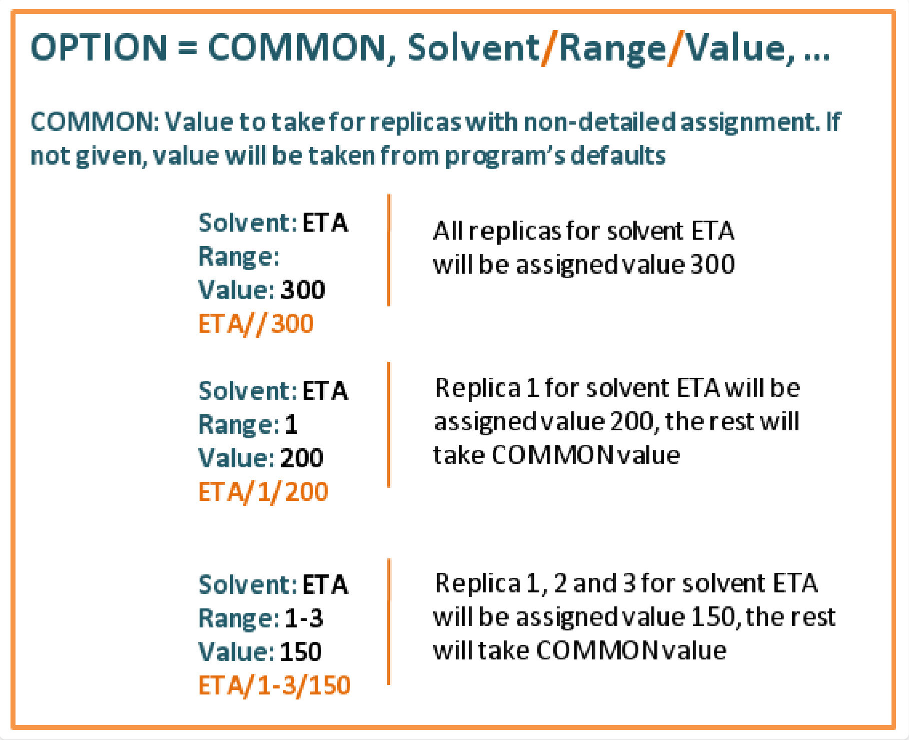
```
[MDSETTINGS1]
solvents = ETA, WAT
nrepl = 3, WAT:1
```

A list of comma separated pairs SOLVENT:NREPL constructions will identify which solvents are to be replicated NREPL times. The first number will apply to all solvents not particularly defined. In this example, any solvent identifier in solvents option will be replicated 3 times but for WAT solvent which will be replicated only once. In this same manner, we could differentially identify any solvent list we could imagine:

```
[MDSETTINGS1]
solvents = ETA, WAT, MAM, ISO
nrepl = 3, WAT:1, ISO:5
```

Here we would create 3 replicas for ETA and MAM mixture, 1 for WAT mixture and 5 for ISO mixture.

Recalling last example in previous section: we wanted to simulate 1 replica of ETA using restraints for 50ns and 1 for WAT for 10ns without restraints. In a similar way we can declare independent number of replicas for each solvent, it is also possible to declare independent restraining modes (RESTR option), nanoseconds (NANOS option), temperatures (TEMP) and forces (FORCE) using a complex syntax. In this case, it is not enough to declare the solvent but also the replica number. This diagram shows how this syntax work:



OPTION = COMMON, Solvent/Range/Value, ...

COMMON: Value to take for replicas with non-detailed assignment. If not given, value will be taken from program's defaults

Solvent: ETA
Range:
Value: 300
ETA//300

All replicas for solvent ETA will be assigned value 300

Solvent: ETA
Range: 1
Value: 200
ETA/1/200

Replica 1 for solvent ETA will be assigned value 200, the rest will take COMMON value

Solvent: ETA
Range: 1-3
Value: 150
ETA/1-3/150

Replica 1, 2 and 3 for solvent ETA will be assigned value 150, the rest will take COMMON value

This is easier to explain and understand with some examples:

```
[MDSETTINGS1]
solvents = ETA, WAT
nanos = ETA//50, WAT//10
restr = ETA//HA
force = 0, ETA//0.5
```

We have declared two solvents (ETA and WAT). One replica for each of them will be created (no NREPL option declared and default value is 1). All replicas with ETA solvent will be assigned 50 nanoseconds and all replicas with WAT solvent will be assigned 10 nanoseconds. In this example, there is no COMMON value and it will be taken from defaults (although there are no more replicas left to be declared). In the same manner RESTR and FORCE options are declared: all replicas for solvent ETA will be restrained over the heavy atoms (HA) using a force constant of 0.5 kcal/mol·A². Water replicas are not declared and will take defaults.

Let's get it a bit more complex:

```
[MDSETTINGS1]
solvents = ETA, WAT
nrepl = ETA:6, WAT:1
nanos = 20
restr = HA, ETA/1-3/FREE
force = 0.5, ETA/1-3/0.0
```

Here we will create 6 replicas for ETA mixture and 1 for WAT which will run for 20ns each. The first 3 replicas of ETA (1-3) will be simulated without restrains. All the other replicas (WAT and ETA 4 to 6) will have restrained atomic positions with a force of 0.5 kcal/mol·A².

With this same syntax one can define different temperatures as well (3 replicas at 298K and 3 more at 400K):

```
[MDSETTINGS1]
solvents = ETA
nrepl = ETA:6
nanos = 20
temp = ETA/1-3/298, ETA/4-6/400
```

MSC File template

```
# pyMDMix MD Settings Configuration File (MSC File)
# All non used options should be commented/removed
[MDSETTINGS]
###########
# GENERAL #
###########
# Comma separated list of solvent box names to be used in the project (e.g. MAM,E
SOLVENTS =
# Number of replicas for each solvent (read documentation for advanced options).
#NREPL =
# Number of nanoseconds to run (advanced configuration in the documentation). E.g
#NANOS =
# Temperature of each replica (again advanced configuration assigning independent
#TEMP =
# Restraining scheme (OPTIONS: HA, FREE, BB for heavyatoms, no-restraints, back-b
#RESTR=
# Restraining force when RESTR!=FREE. In kcal/mol.A^2. Default=0.0. Advanced conf
#FORCE=
# Residue mask of residues where restraints should be applied (default: auto dete
#RESTRMASK = auto
# Residue mask of residues to which backbone atoms we should align the trajectori
#ALIGNMASK = auto
#####################
# SETTINGS OVERRIDE #
#####################
# All parameters present in replica-settings.cfg can be here overriden.
# For instance, if we want to modify the trajectory writing frequency:
# TRAJFREQUENCY =
# Modify number of steps per run file (step).
# We might be interested in increase the file size to include several nanosecods
# instead of only 1 (the default; 500000 steps with md_timestep of 2fs)
# PROD_STEPS = 1000000
# NPT_PROD_STEPS = 1000000
# ETC...
```

Default MDMix simulation protocol

When building the project, input files to run the simulations will be automatically generated. The default simulation protocol implemented in the package is the following:

1. *Minimization*: 5000 steps.
2. *Equilibration*: Heating from 100K to the final temperature in 800ps in NVT ensemble + 1ns of NPT equilibration at the final temperature. Langevin thermostat. 2fs timestep.
3. *Production*: NVT ensemble with Langevin Thermostat, PME for electrostatic calculations. Timestep of 2fs. Each file will correspond to 1ns of simulation: writing frequency of 500 and number of steps of 500.000.

Package/User Default MD settings

pyMDMix is distributed with a file which defines all default MD parameters all projects and users will be using if no other instructions are given. This file is located at the python module installation directory (**$PYTHON_INSTALL_DIR/pyMDMix/data/defaults/md-settings.cfg**) and will be copied to the user's home directory at first execution as well (**$HOME/.mdmix/md-settings.cfg**). The file will be similar to this one:

```
[GENERAL]
## The following options are only ckecked for their type
## use type-name if you want to enforce type conversion on a parameter
## example:
## int-param1 = 10 creates a variable param1 of type int with value 10
## By contrast, param1 = 10 gives a variable param1 of type str with value '10'
## Possible types: int, float, bool, list.
## list type will chop the string by commas. Eg. list-ff=a,b will become a list f
# Set simulation options
 mdnetcdf = 1 # 1 (write trajectory in nc format) or 0 (write in ascii format)
 restrMode = FREE # Restraining scheme: FREE, HA (heavy atoms) or BB (backbone on
 float-restrForce = 0.0 # Restraining force if applicable. Default 0 kcal/mol.A^2
 int-nanos = 20 # Production length in nanoseconds. Default: 20ns
 float-temp = 300 # Simulation temperature. Default = 300K
 mdProgram = AMBER # Default simulation program. Options: AMBER or NAMD currently
 int-trajfrequency = 500 # Trajectory writing frequency = 1000 snapshots per nano
 int-minsteps = 5000 # Number of minimization steps to run
 int-heating_steps = 100000 # Heating steps for each file. 100.000 steps = 200ps
 float-parm_heating_tempi = 100 # Start heating at 100 K
 int-npt_eq_steps = 500000 # 1ns equilibration at NPT
 int-prod_steps = 500000 # 1ns production files = nvt_prod_steps*(md_timestep/10e
 float-md_timestep = 2 # 2 fs timestep
 int-namd_heating_steps = 500000 # 1ns equilibration total time to increase tempe
 list-FF = leaprc.ff99SB, leaprc.gaff # Default forcefield files to load when ope

# DEFINE REFERENCE STRUCTURE FOR RESTRAINED SIMULATIONS
# If 1: Use output of minimization as reference structure to set positional restr
# If 0: (default) use initial input structure (CRD file) as reference state.
 int-minimizationAsRef = 0
```

As you may probably have realized, some of the options are prefixed with float- int- or list-. These prefix will not be part of the parameter and internally identify the value type. Be careful to keep these prefix if the parameter should be modified.