pyMDMix

The python package for your organic mixtures simulations

Solvent mixtures

Creation and management of solvent mixtures is a key point of MDMix methodology. pyMDMix currently incorporates a solvent database with a few already prepared solvent mixtures described here below. It also incorporates a solvent management system that allows the user to easily extend the solvent library and identify probe atoms and chemical types of those probes for the new solvent mixture.

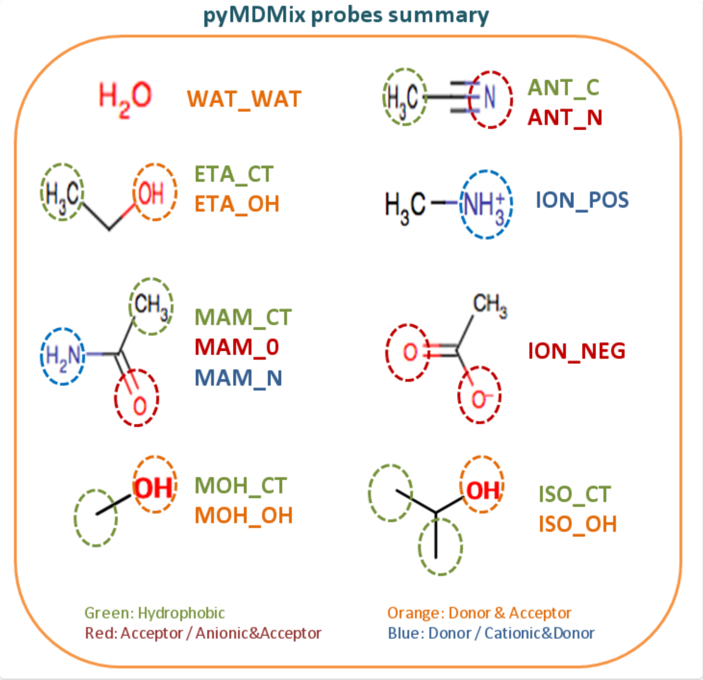
pyMDMix package solvent mixtures

At any time, you can query the program for available solvent mixtures issuing the following command. These are the distributed solvent mixtures which are ready to use:

```
[dalvarez@nivil-2:~]>mdmix info solvents
--------------------------------------------------
SOLVENT DB in USE: /Users/dalvarez/Dropbox/WORK/pyMDMix/pyMDMix/data/solventlib/SOLVENTS.db
Package SOLVENT DB: /Users/dalvarez/Dropbox/WORK/pyMDMix/pyMDMix/data/solventlib/SOLVENTS.db
SOLVENTS Directory: /Users/dalvarez/Dropbox/WORK/pyMDMix/pyMDMix/data/solventlib
Solvents:
 SOLVENT: MOH    INFO: Methanol 20% mixture      BOXUNIT: MOHWAT20       PROBES: MOH_WAT, MOH_OH, MOH_CT
 SOLVENT: ANT    INFO: Acetonitrile 20% mixture  BOXUNIT: ANTWAT20       PROBES: ANT_C, ANT_WAT, ANT_N
 SOLVENT: ION    INFO: Ammonium Acetate mixture (Charged mixture)        BOXUNIT: CN3COOWAT      PROBES:
 SOLVENT: ETA    INFO: Ethanol 20% mixture       BOXUNIT: ETAWAT20       PROBES: ETA_CT, ETA_WAT, ETA_OH
 SOLVENT: ISO    INFO: Isopropanol 20% mixture   BOXUNIT: ISOWAT20       PROBES: ISO_WAT, ISO_OH, ISO_CT
 SOLVENT: ISO5   INFO: Isopropanol 5% mixture    BOXUNIT: ISOWAT5        PROBES: ISO5_WAT, ISO5_OH, ISO5_
 SOLVENT: WAT    INFO: Water box BOXUNIT: TIP3PBOX       PROBES: WAT_WAT
 SOLVENT: MAM    INFO: Acetamide 20% mixture     BOXUNIT: MAMWAT20       PROBES: MAM_N, MAM_WAT, MAM_O,
--------------------------------------------------
```

Probes summary

This picture describes the probes already distributed with the package solvent database. Moreover, if the mixture contains water, a water probe for each mixture is also included (see description above, e.g. ETA_WAT means the water probe of the ETA mixture).



Creating a new solvent mixture

This process is not currently implemented in pyMDMix. Although it is planned to be included in future releases, at the moment the user is in charge of manually creating his/her specific solvent mixtures. The workflow might be something like this:

1. **Parameterize small molecules included in the mixture:** Small organic molecules should be parameterized using amber forcefields ensuring correct geometries are obtained.
2. **Build the the mixture box:** Using packmol or any other software, construct the box with needed number of water and co-solvent molecules to make up the approximate desired concentration.
3. **Equilibrate the box:** Run a molecular dynamics simulation to equilibrate the box volume and velocities to 300K and 1 atomsphere pressure. Run long enough to obtain a stable system.
4. **Create the box with Ambertools:** Last step is to load the box in s/x/tLeap program from Ambertools package and create a new unit with the output of the equilibration stage. This new unit should be saved as an Amber Object File (*.off or *.lib extensions usually). This OFF file should contain the equilibrated box and also individual residues that make up the box along with their tailored parameters if any.

A tutorial with a practical step-by-step guide is planned to be incorporated to the website as soon as possible.

Adding a new solvent mixture to pyMDMix

This process involves the creation of a **Solvent Configuration File (SCF)** which will tell the program how to treat the solvent. The file contains three sections:

1. **General:** Information about the box unit in the OFF file.
2. **Probes:** Map user defined probe names to atoms in the solvent box. This section will tell pyMDMix what atoms should be tracked down during the simulation analysis to build up the energy grids. These atoms will have a unique probe name to identify them.
3. **Types:** Assign chemical types to the probes previously defined. The type names are up to the user. These names will be used for joint analysis of probes across different solvents (e.g. if two solvent boxes were simulated and both contain donor probes, it will be possible to join them).

This file was used to add the ethanol/water mixture to pyMDMix and can be used as template for any other solvent mixture. Pay attention at the options inside each section:

```
[GENERAL]
# name to identify the mixture (ex: ION)
name = ETA
# Any string to describe the box (Use %% to scape % sign)
info = Ethanol 20%% mixture
# path to off file containing the leap units
# It should contain all parameters
objectfile = ETAWAT20.off
# If the box contains waters, the name of the model (TIP3P, TIP4P, SCP..)
```

```
watermodel = TIP3P
# Name of the Leap box unit in object file(ex: IONWAT20)
boxunit = ETAWAT20
[PROBES]
# OPTIONAL SECTION
# map probe names with residue@atoms (ie. NEG=COO@O1,O2)
# probe names must be unique
WAT=WAT@O
CT=ETA@C1
OH=ETA@O1
[TYPES]
# OPTIONAL
# Assign chemical types to the probes in previous section
# Example: OH=Donor,Acceptor
OH=Don,Acc
CT=Hyd
WAT=Wat
```

GENERAL section describes the box:

- **name:** Identification name of the solvent mixture.
- **info:** A short string describing the mixture.
- **objectfile:** Path to the OFF file saved using tLeap in the box creation process.
- **boxunit:** Name given to the unit containing the pre-equilibrated box inside the OFF file. Remember to add other units with each residue present in the mixture (including the water!) and any extra parameters needed.
- **watermodel:** Water model used for the mixture (e.g. TIP3P, TIP4P, SCP, …). If using a non standard type, include the parameters in the object file and remove this entry.

PROBES section identifies which atoms to track:

> Here the user may add as many entries as needed. For this particular example, we are interested in tracking the water (O atom name in residue name WAT), the ethanol oxygen (O1 atom name in residue name ETA) and the carbon from the methyl tail (C1 atom name of residue name ETA). The line is constructed following this template:

```
PROBENAME = RESNAME@ATOMNAMES
```

> Being *ATOMNAMES* a single name or a list of comma separated names (e.g. in ION box mixture, NEG probe is built from O1 and O2 atoms in COO residue):

```
NEG = COO@O1,O2 # Example from ION box configuration file
```

> If you were interested in tracking also the carbon atom bound to the oxygen (C2 atom in this example), we could add a line as the following to this section:

```
CT2=ETA@C2
```

TYPES section:

> Here we will need as many lines as probes defined in the previous section. After equal sign, a list of type names can be provided. In ethanol mixture, we are identifying *OH* probe as Donor or Acceptor (*Don,Acc*). Being Don and Acc type names that can be changed to any desired type (but be consistent when more solvents are added to the database!).

Database management

A database is distributed with the package and will be installed under the python package root directory. If the package installation is performed by a user with administration privileges, a normal user would not be allowed to modify this solvent database. In this situation, the program will automatically copy the package database into the user’s home directory.

A folder **.mdmix/** will be created and the database saved inside with name **SOLVENTS.db**. Now the user will be able to modify it and add any required solvent mixture.