Getting started

pyMDMix allows an easy set up of several simulations for the same system under different conditions: solvent, temperature, restraining schemes, etc. Moreover, after simulations are done, many analysis tools will help you to quality check and extract useful information from these simulations in a aqueous-organic environments.

Project setup up workflow

1. Manual system preparation. Generate an AMBER object file with Leap (or PDB file) which is the main input to pyMDMix. Check out System preparation.
2. Create an empty pyMDMix project and add the system to study.
3. **Create replicas of the system**: A replica is considered as a single independent MD system and will contain all input files needed to run the simulation (including commands to be executed and queue input files if requested). Chose solvent and simulation conditions (check MD settings for details).
4. **Run the simulations**. This is done by the user in his/her computation facility. The execution commands may be in this point adapted to your cluster specifications (i.e. modify queue input files and so on). This step is out of control of the program as every cluster has specific operation rules.
5. Bring back results to the project and start analysis.
6. **Analyze results!** Common analysis involves the generation of density maps for each solvent and probe previous alignment of the simulation trajectory. These maps are transformed to free energies. Check out Analysis Guide for more information and details.

pyMDMix User Interface

The software, beyond a python package, is distributed with a command line user interface that allows the execution of most common operations for a complete MDMix setup and analysis study. Once installed, the program is executed by calling `mdmix` from the command line with the following syntax:

```
> mdmix [options] {tasks}
```

Where `options` are listed in the last part of help message (basically –log or –debug) and `tasks` are listed under positional arguments in the help message:

```
> mdmix --help
 ============================================================
 ||          pyMDMix User Interface                        ||
 ============================================================
 ||    Author: Daniel Alvarez-Garcia                       ||
 ||    Version : 0.1                                       ||
 ============================================================
usage: mdmix [-h] [--log LOGFILE] [--debug]
                          {create,info,add,remove,queue,plot,analyze,tools} ...
positional arguments:
    {create,info,add,remove,queue,plot,analyze,tools}
                        commands
     create             Create Project or Solvents
     info               Print information about the project or solventDB
     add                Add new replicas, systems or create group of replicas
                        in an existing project.
     remove             Remove groups from project. To remove systems or
                        replicas, simply remove the forlders or system files
                        from the project folder.
     queue              Queue input files options.
```

```
    plot              Plotting command
    analyze           Several analysis tools to run on the replicas.
    tools             Complementary tools
optional arguments:
    -h, --help        show this help message and exit
    --log LOGFILE     Logging file. Default: output to stdout
    --debug           Print debugging info
```

Each task will have its own options and the full command picture will be like this:

```
> mdmix [options] {task} [task-options]
```

Where task specific options can be found calling the command help and are explained in detail in the corresponding documentation sections.

```
> mdmix [options] {task} --help
```

Some examples

```
> mdmix --debug plot rmsd
```

Here the output will be verbose for debugging purposes. The task in this example would be `plot` and `rmsd` is a task specific option.

```
> mdmix --log align.log analyze align all
```

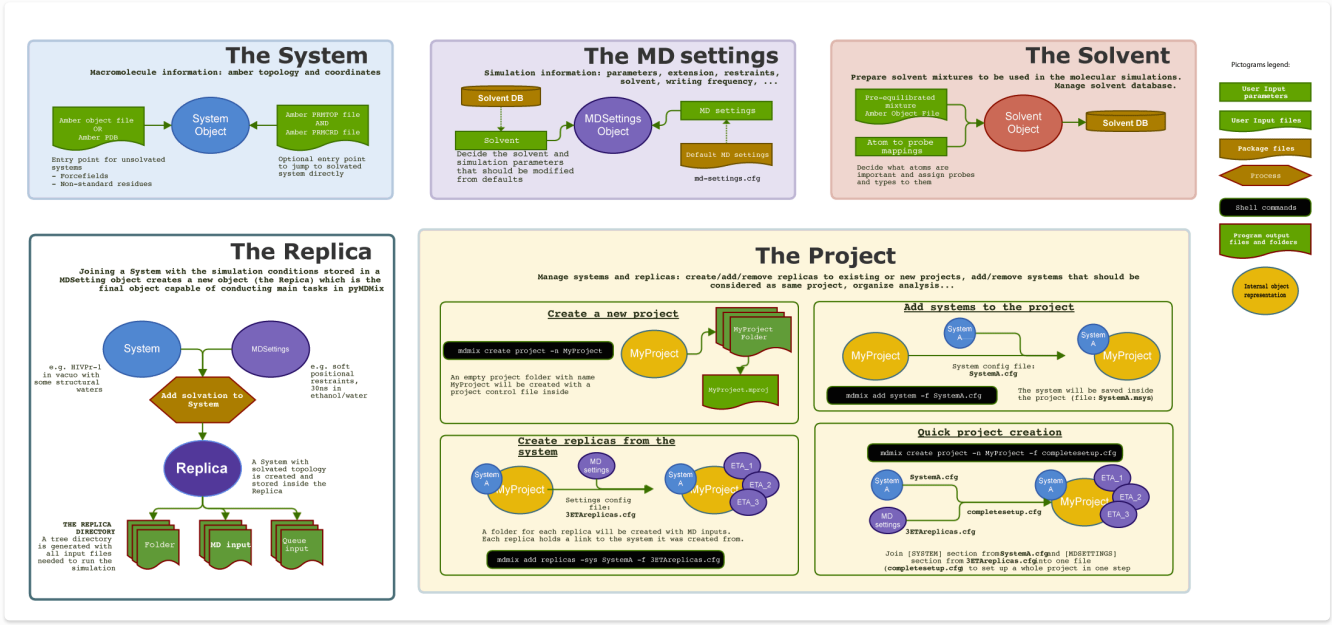
Here all the program output will be redirected to the logging file `align.log`. Task is `analyze` and `align all` are task specific options.

Introductory tutorials

Follow the introductory tutorials to help you kick off your first projects.

pyMDMix diagram

To better understand the program operation, this picture illustrates some of the main objects the program internally manages and how the user interacts with them through a pyMDMix project. This graphic will help you understand many parts of the documentation:



- **The System:** Macromolecule container. Usually configured from mdmix user interface with a system configuration file. Check System section. Two main ways of building a system exists: with an amber object file or a amber-ready PDB.
- **The MDSettings:** Configuration parameters for molecular dynamics simulations. Configured with default parameters in replica-settings.cfg and user customizable when creating replicas of a system. Check MDSettings and Replicas sections.

- **The Solvent:** Object containing the solvent mixtures boxes that can be loaded into tLeap for solvating the macromolecule. The program includes a bunch of pre-equilibrated and ready to use solvent mixtures. Check Solvent Sections.
- **The Replica:** When combining a System with some MD Settings, a folder structure with MD inputs is created inside the project. Generation of replicas for a system is done through a replica configuration file. Each replica is an independent entity.
- **The project:** Internal object to manage replicas and systems.