Creating Replicas

A Replica is considered a single independent MD setup. During replica preparation, the unsolvated system is placed inside the solvent mixture chosen and topology and coordinate files are writen inside the replica folder. According to the MD settings chosen, input configuration files for running the simulation are also writen. The user can move this replica folder to the computation facility and return it to the project folder whenever the simulation is finished. The replica will identify if the simulation is completed and will orchestrate analysis. (Visit Getting started pictogram for a visual summary of what a Replica is).

Internal operation

The creation process of a replica consists of the following steps:

1. System solvation**:** The system and the chosen solvent mixture are loaded into tLeap using object files (the system generated object file and the object file stored in the solvent database). The following commands are called inside tLeap:
   - *solvateOct*: Using the pre-equilibrated solvent box and a buffer specified in program defaults.
   - *addIons*: Neutralize the system with NaCl counterions. If the pre-equilibrated box is a binary charged mixture (e.g. ION box), the sytem is neutralized removing the necessary charged solvent residues: if a total charge of +3 is present after solvations, 3 positively charged residues from the solvent mixture will be removed to neutralize.
   - *saveAmberParm*: PARMTOP and PARMCRD will be saved and stored inside the replica object.
2. Replica folder creation: Write to disk a folder tree containing PARMTOP and PARMCRD as well as all MD configuration files. Also a PDB file from the topology and coordinates will be saved (ambpdb from AmberTools is called).
3. If the replica is set to have restraints, an extra PDB file will be saved which will be used as reference for setting the restraints positions.
4. Writing of simulation input files and commands for NAMD or AMBER software.

Replica creation

For creating a simulation replica, pyMDMix needs to know two things:

1. The system we want to simulate: A System should have been prepared beforehand and added to an existing project.
2. The simulation settings: solvent and MD conditions will define individual replicas. Definition of these settings is done through the **MD Settings Configuration File** (MSCF). The file might contain one set up for defining one single replica or might be configured to contain multiple settings for creating multiple replicas at once. Find a description and examples in MD Settings section.

Linking systems with settings results in individual MD set ups (the Replicas). To do so, we must add replicas to the current project. If our MSCF file is called **`mysettings.cfg`**, and we have added a System to the project called **`SystemA`**, the command to be executed would be:

```
> mdmix add replicas -f mysettings.cfg -sys SystemA
```

This will create one folder for each of the replicas identified by the different md setups in `mysettings.cfg` (the Replica folders).

Replica Folder

The folder created will contain several subfolders and files:

Files:

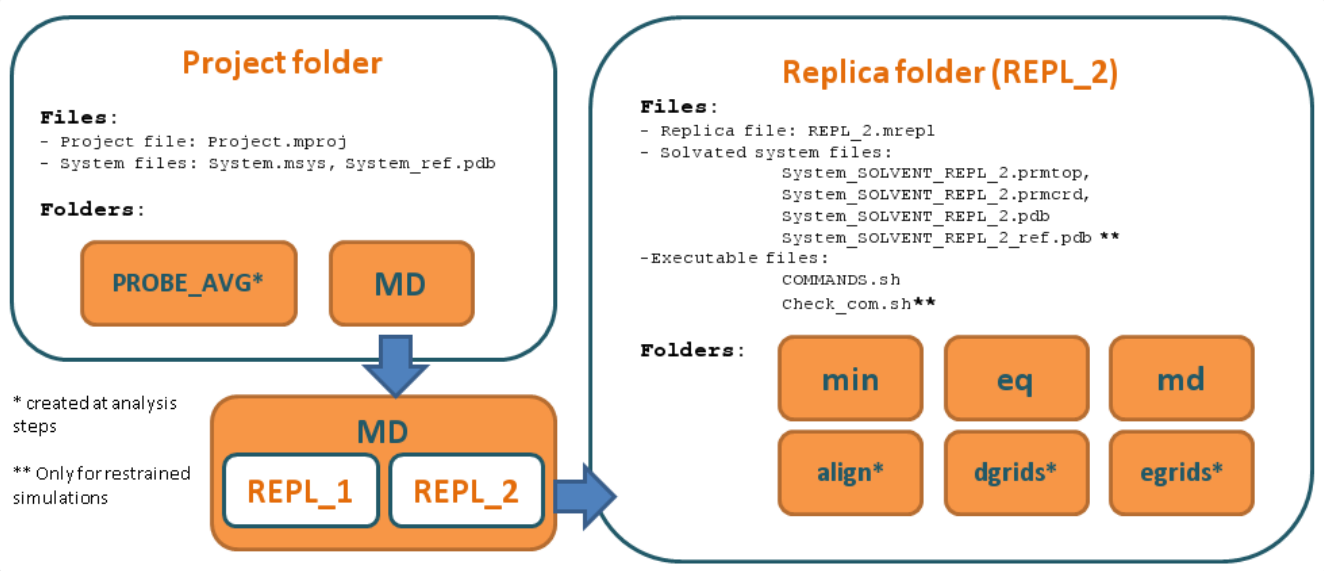
- A file with same name as the folder ended with `mrepl` which is internally used by pyMDMix to identify the folder and contains all replica parameters. This file should not be removed!
- Files with the topology and coordinates of the solvated system (system in study after adding the solvent mixture). Also some PDB files will be created so the user can visualize the solvated system. More over, a file with name _ref.pdb will be used as reference for trajectory alignment during analysis stages.
- An executable shell script COMMANDS.sh contains all commands that the user should execute to run the simulation in a serial manner. This file is here as a template for the user to know the expected output file names the program will seek during analysis process.

- If the simulation uses restraints, an extra shell script called check_com.sh will be writen. This script will be called after each simulation step to center and re-align the protein structure over the reference structure.

Folders:

- **min, eq, md**: contain input configuration files for running the simulation. Correspond to minimization, equilibration and production stages.
- **align**: created during trajectory analysis, will contain ptraj input files and resulting aligned trajectory plus rmsd calculations and protein averages for each step (see Analysis Guide).
- **dgrids, egrids**: if required, these folders will be created at analysis stages. **dgrids** will contain density grids for each probe in the simulated solvent and **egrids** will contain the energy converted grids.

Folder diagram:



Creating groups of replicas

During analysis process you might want to do joint analysis of a group of replicas that share same simulation conditions. For instance, if we have simulated a system with ethanol using restraints and without restraints, we will be interested in analyzing these two groups separately. To do so, we could create two groups:

```
> mdmix add group -gn ethanol_restrained -s ETA_1 ETA_2 ETA_3
> mdmix add group -gn ethanol_free -s ETA_4 ETA_5 ETA_6
```

We have created `ethanol_restrained` and `ethanol_free` groups. The first one is composed by `ETA_1` to `ETA_3` and the second from `ETA_4` to `ETA_6` replica names. Check Analysis Guide section for details on how to use this groups for identifying replicas during analysis.