Molecular System Preparation Scripts

Latest vers. 18/06/2023

Introduction

This an ensemble of scripts designed to facilitate and speed up the setup of molecular dynamics simulations of small molecules, proteins and protein - ligand complexes. The scripts were written to work with the Gromacs simulation package and as such are probably only useful if gromacs is already installed. The scripts are high level so it doesn't matter which version of gromacs you're running providing it's > 5.x.

Installation

To get started, clone the repository:

```
$ git clone https://github.com/gpieffet/msps.git
```

Then the easiest is probably to set your **PATH** to include the newly created directory *msps* containing the scripts:

```
$ cd msps
$ pwd                                                         # copy the resulting p
ath
$ export PATH="$PATH:path_to_msps"
```

Setup

When using the scripts, you'll first have to set the variables ROOTDIR, FF_DIR, DOCK_DIR and DOCK_DIR_GLIDE inside the file prep_VAR. The file prep_VAR is parsed by many of the scripts to know where to find the structures and trajectories that need to be processed, among other things:

- ROOTDIR: the root directory of the project
- FF_DIR: the location of the force field (FF), only needed if using a FF not included in the default distribution
- DOCK_DIR: the location of the docking results from vina
- DOCK_DIR_GLIDE: the location of the docking results from Glide

Requirements

Appart from gromacs, the following programs are also needed by some scripts:

- babel/obabel for file format conversion
  - note: vers. 2.4 is advised, as from vers. 3.0 and up flags have changed and obabel desn't seem to be able to consistently generate reasonable protonated structures
- vina and vina_split for running docking simulations
- antechamber for generating ligand topology files (including RESP charges from QM calculations)
- gmxMMPBSA for MM/P(G)BSA calculations

Note that the scripts expect the result files to be organized and named in a certain way, i.e. the docking results for the protein PROT are expectd to be located in the directory named PROT and the name of the docking results of the ligand LIG with PROT should be PROT_LIG-run.
Clustering the docking results is done using the script *prep_cluster*:

```
$ prep_cluster
Syntax: prep_cluster protein ligand [nb_cluster]

$ prep_cluster PROT LIG 4
```

Where PROT and LIG correspond to the protein and ligand names, respectively.
The script should generate a file PROT_LIG.pdb containing the structures of the center of 4 clusters accross all the docking poses.

Workflow

After preparing the ligand (for instance with gaussview) and the protein (for instance with maestro), a sequence of actions could be:

1/ Generate the pdbqt files with adt if using vina (you'll have to do it manually) and run the docking simulations.

```
$ vinad PROT.pdbqt LIG.pdbqt param.conf
```

The interest of using the script *vinad* instead of vina directly is that it allows you to run multiple docking simulations with systematic naming of the result files. Running multiple docking simulations can be useful since vina uses a stochastic approach.

2/ Sort the docking results into a specific number of clusters and select the center of each cluster

```
$ prep_cluster PROT LIG.pdb NB_CLUSTER
```

The script *prep_cluster* calls the script *vina_aggregate*, *obabel* and *gromacs (cluster)* to organize, convert and cluster the docking results. You need to indicate how many clusters you want and the script will adjust the initial cluster cut off value (the default cutoff 0.40 nm is a value that seemed to work well for many medium size ligands) until the desired number of clusters is reached. If no convergence is reached, change the cutoff time step DT inside the script to a lower value.

3/ Calculate the (ESP) charges of the ligand (for instance with gaussian) based on one of the docking poses (no scripts to help with this step, you'll have to do it manually)

4/ Generate the ligand topology with antechamber using the calculated (ESP) charges

```
$ prep_top_lig LIG.log
```

5/ Run the MD simulation of the ligand (min, eq and producion run)

```
$ prep_sim_lig LIG
```

6/ Prepare the topology of the protein

```
$ prep_top_prot PROT
```

7/ Run the MD simulation of the protein

```
$ prep_sim_prot PROT GPU_ID
```

8/ Prepare structure and topology of the protein-ligand complex using the first cluster center cc0

```
$ prep_top_compl PROT LIG cc0
```

9/ Run the MD simulations of the complex (with MM/PBSA or MM/GBSA calculations)

```
$ prep_sim_compl PROT LIG cc0 GPU_ID
```

Run the simulation of the first cluster center cc0 of the protein - ligand complex. Specify the GPU id number of the card you want to run on. Obviously you'll need to have gromacs compiled with GPU support to do that.

Examples

The cloned directory *msps* comes with an example to determine the cluster centers of docking results of the so-called ligand 7C with the progesterone receptor (PDB 1osh).

If you haven't done it already, add the path up to and including the directory *msps* to your variable **PATH**:

```
$ cd msps
$ pwd
$ export PATH="$PATH:path_of_msps"
```

The directory *examples/docking* contains the docking results of ligC with 1osh, therefore we set the variable **DOCK_DIR** with that path in the file prep_VAR:

```
$
$ echo "DOCK_DIR=`pwd`/examples/docking" > prep_VAR
```

Note that that the file *prep_VAR* must be inside the directory *msps*.

```
$ cd examples/docking
$ ls 1osh
```

The directory *1osh* inside *examples* contains the results of 5 docking simulations, each with 9 poses.

We request the classification of the all the *5 x 9 = 45* conformations of ligC into 4 clusters:

```
$ prep_cluster 1osh ligC 4
```

The center structures of each of the 4 clusters are written to the file *1osh_ligC.pdb* and you can now use these ligand conformations as starting structure for the MD simulations of your protein - ligand complex.