Polyatomic correction

The polyatomic correction of the energy grids obtained by MDMix tries to correct the efect that one probe has on the ohter(s). This phenomena is known by everybody that has done MDMix, as most hotspots appear by pairs (i.e. one polar and one hydrofobic) due to their nature (they are tied by a bound). To correct this, the polyatomic correction extracts the energy from the whole molecule and repartitions it in the probes, depending on the impact each one has on the global affinity of the solvent.

1. Dependecies

Similar to the hotspot clustering, the original script and algorithm was writen by Daniel Alvarez and use cryptic and un-mantained libraries. This is specially true for the polyatomic correction, as it uses functions that are incorporated to libraries in MDMix (most of them either to biskit or the pyMDMix library). In that regard, I have only been able to adapt the script to function under the python environment of MDMix (in BSC). Thus, everything explained here will be suposing you have access to that environment and are using the MARENOSTRUM service.

2. Files and formats

Even though the script uses MDMix functions to work, we will need to adapt many of the files given by MDMix for the script to run. The correct_poly_new.py (main script) only accepts one input file: a configuration. The file needs to be in the following format (I will leave and example in the ./example folder):

```
 [OPTIONS]
solvent = SOLVENT
mdprog = amber
outpath = /PATH/TO/STORE/THE/OUTGRIDS/
trajpath = /PATH/TO/THE/ALIGNED/TRAJECTORIES
syspdb = /PATH/TO/THE/REFPDB/WITH/ALL/ATOMS/example_ref_allatoms.pdb
ext = crd

[GRIDS]
CT = /PATH/TO/GRIDS/ETA_CT_DG0.dx
OH = /PATH/TO/GRIDS/ETA_OH_DG0.dx
CM = /PATH/TO/GRIDS/ETA_COM_DG0.dx
```

In the *solvent* you need to put the name of your solvent as you put it on the MDMix configuration file (i.e. ETA, MAM, etc.). Leave the *mdprog* as amber unless you are using namd, in which case good luck. The *outpath* and *trajpath* are the folders where you want your files to be stored and where you have your trajectories respectively. Make sure the *syspdb* coincides with the trajectories (its the same as the topology) as it will crash otherwise. The *ext* is where we come to the first format issue: if your MDMix uses AMBER, by default it stores the trajectories in the NetCDF format (*.nc), but the script only accepts *.dcd or *.crd. In order to perform the polyatomic correction, you will need to transform all your trajectories to the crd format (its occupies about double ~ double the size than NetCDF). I used cpptraj to do it, but any tool (mdtraj, pytraj, etc) should work. I leave the queueing file and the cpptraj input template in the ./lib/nc2crd folder (you will need to change the topology in the template).

The GRID files, are the normal energy grid files you generated with MDMix, with the exception of the **CM** which refers to the center of mass. If you are already using the cpptraj_density way of calculating the density grids for MDMix, you will only need to apply the grid command in cpptraj using the atom you defined as the center of mass (e.g. C1 for ETA) or match the mask to the whole solvent and specify to the script that it uses the center of mass. It should be something similar to this :

```
 parm mysystem.prmtop
trajin align/md*nc
grid CENTER_OF_MAS.dx 200 0.5 200 0.5 200 0.5 gridcenter 1 1 1 :ETA byres
run
```

There are many more things you can do with the grid command in cpptraj (setting up thresholds, creating a pdb with the best spots, etc), check their manual. You are very welcome to suggest improvements (as we have only used the most basic functionality)

3. Usage

As said before, the script only needs the configure file, so launching it is pretty simple. Nonetheless I recomend you to launch it using the queuing system using a queue file similar to this

```
 #!/bin/bash
# @ job_name = analysis
# @ initialdir = .
# @ wall_clock_limit = 2:00:00
# @ output = polyatomic_correction.q.o
# @ error =  polyatomic_correction.q.e
# @ total_tasks = 1
# @ tasks_per_node = 1
# @ cpus_per_task = 1

# marenostrum
# OJO load on your enviroment too!!
source /gpfs/projects/ub63/apps/python/bin/activate
module purge
module load intel impi mkl netcdf hdf5 amber

python lib/correct_poly_new.py polyatomic_corr.conf
```