Instructions to the dRama script

- The program creates what we call a drama plot. That is, it calculates as a 2d histogram of rotational angles around alpha carbon.
- Data are taken from a text file generated by Gromacs.
- Results are saved in various formats from colour-mapped plots to data in text files.

Input data

The program takes two data files, one of them called: “system” and the other: “reference”. They both contain the values of psi, and phi angles formatted in the same way, that is like a Gromacs output file. The file should contain two rows of angular data in degrees and look like the example below:

```
# This file was created Wed Oct 26 14:12:32 2022
…. . .
@    yaxis  tick major 60
@    yaxis  tick minor 30
@ s0 symbol 2
@ s0 symbol size 0.4
@ s0 symbol fill 1
-149.064  142.057  VAL-12
-76.7526  113.545  HIS-13
-145.872  176.683  HIS-14
-59.8088  111.048  GLN-15
74.2803  12.421  LYS-16
-127.593  113.773  LEU-17
-92.7932  124.958  VAL-18
```

All the lines beginning with @ or # are ignored. For the other lines, only the first 2 columns are loaded.

Requirements

The script requires at least Python 3.2 and a *numpy* module installed. To install *numpy*, install Python first and then type into the command line the following

```
pip install numpy
```

of

```
pip3 install numpy
```

if you are using Linux. At Linux, you might have to install pip as well.

Additional Libraries

With only the numpy module installed the program will only output the data in text format. To be able to obtain graphs and pictures the tifffile and matlibplot need to be installed. To do so type into the command line the following commands.

```
pip (or pip3) install tifffile
```

```
pip (or pip3) install matplotlib
```

If you are using the drama script from the command line you also need to install argparse

```
pip (or pip3) install argparse
```

Usage

Put your data and drama.py script in the same directory. Then in the command line cd to that directory and type in:

```
python (or python3) drama fileS.xvg fileR.xvg
```

where fileS.xvg fileR.xvg are names of input files.

Parameters

The first two parameters of the function are obligatory, they are the names of the input files. There are a few optional named parameters coming in two types: boolean and numeric or string. They differ in usage, for the numeric and string parameters the usage is the following:

```
python (or python3) drama fileS.xvg fileR.xvg --binSize 1.5
```

The above command sets the parameter binSize to the value of 1.5

for boolean parameters, you just type in the parameter name with no value to set it to value "true" like in the following example:

```
python (or python3) drama fileS.xvg fileR.xvg --pureText
```

The above command sets the parameter pureText to the value "true". Setting pureText to true is a good option if you do not want to or have a problem instilling the graphical packages. You can use this parameter to get the script to work and obtain only the test output.

Parmeter list:

saveDir

The directory where the files are saved. It is a subdirectory of a current folder, the path should be relative to the current folder, like “my_test_results” not “C:\ my_test_results|”. If not set, the program will create a directory named “results” in the current folder.

SaveName

The file name under which the result will be saved. If not specified, the files will be saved under the first (system) input file.

binSize

The bin size for binning is in degrees. The bigger the number the coarser the mesh.

vMax

the value used for manual scaling of the “density” plot. The higher the value the more saturated the plot.

Borderline

Similar to vMax but for the differential plot. Adjusts the saturation of the plot (lower value - more saturated)

PlotRaw

Plots traditional Ramachandran ploy of raw, non-binned data.

dontSaveAsTif

prevents saving data as tiff file

dontPlotData

prevents saving data as plot jpg

animatedGif

Saves data as an animated griff file. The file is created in the results directory or in the directory specified as saveDir by the user under the name "animation.gif". Each run of the script adds one frame to the animation.gif file. So, the best way to create animation is to run a script multiple times with different data, for example in a loop.

pureText

Disables all plots and only saves ascii data. With this option enabled the script requires only a very popular NumPy library to work. **Caution advised:** This command overwrites the other command associated with plotting data.