USER MANUAL

Valentina Erastova

The University of Edinburgh

Matteo Degiacomi

Durham University

1. REQUIREMENTS AND INSTALLATION

`dynden.py` should be run through terminal and requires Python >3.4. Furthermore, the following Python packages to be installed in your computer:

- numpy
- matplotlib
- MDAnalysis

There are two ways of installing Python. If one distribution is already available in your operating system (type `python` on terminal to verify), packages can be installed with the command:

```
sudo pip install package_name
```

If you are not super-user of your machine, you can install packages in your home directory, and inform Python about the specific location of those packages.

```
pip install package_name --user
# a typical local installation path for Linux with bash terminal, or related
export PYTHONPATH=$PYTHONPATH:~/.lib/local/Python3.6/site-packages
```

If you need to install a new version of Python, the favorite solution should be Anaconda. Download the version of Anaconda suitable for your operating system here:
https://www.anaconda.com/download/

Anaconda will provide you by default with most Python packages. To install new ones, type in terminal:

```
conda install package_name
```

The package MDAnalysis can be also installed with Anaconda but requires an additional option first to inform the `conda` command about where to fetch this package.

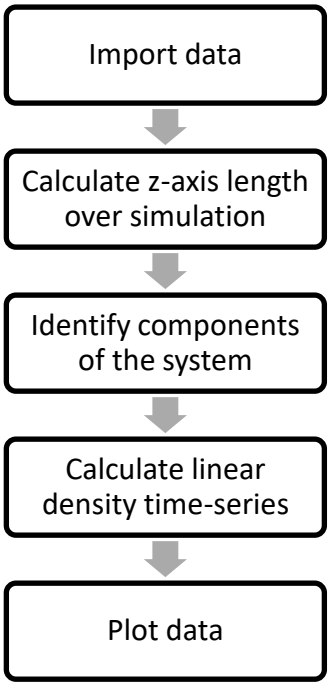
```
conda install -c conda-forge mdanalyis
```

2. OVERVIEW

dynden.py analyzes the evolution of linear densities of individual components through the simulation. The linear densities are calculated orthogonal to the *xy*-plane, so if the simulation features a surface, it should be aligned to *xy*-plane. The analysis works best for simulations NPT ensemble with only minor fluctuations along *z*-axis*.

DynDen pipeline:

1. Import trajectory and topology, read in user-defined parameters.
2. Measure length of the *z*-axis for each frame of the simulation.
3. Identify all the individual components of the system, based on their residue name OR as declared by the user.
4. For each system component and the whole system calculate linear density time series (DTS) and their pairwise correlation coefficients.
5. Generate and save figures.



As the calculation is memory intensive and may take long time, files starting with **bkp_*** are generated and saved in the working directory. These files can be used to speed up regeneration of the data or to replot it. During execution, **dynden.py** will seek for the existence of these files, and perform analysis only if they are not found.

* *z*-axis is defined as orthogonal to *xy*-plane, and not length of a side of the simulation box

3. INPUT PARAMETERS

To run *DynDen* for the first time, use the following call on terminal:

```
dynden.py -t trajectory.trr -s topology.tpr [optional keywords]
```

Optional keywords are:

`-t` trajectory file, please refer to the section 6.1 of the MDAnalysis manual for information about supported trajectory formats*

`-s` topology file, please refer to the section 5.1 of the MDAnalysis manual for information about supported topology formats*

`-b` number of bins for *z*-axis binning (default: 100)

`-nf` number of time frames to account for averaging (default: 50)

`-am` method for averaging ("block", "slide" or "none", default: "block")

`-f` first frame of simulation to include in analysis (default: 0)

`-l` last frame to include in simulation (default: number of frames in simulation)

`-ts` frequency of trajectory in picoseconds (default: 10 ps) used to correct plots to report correct simulation time.

`-disp` display plots on screen ("yes" or "no", default: "yes").

`-string "string, separated by commas"` if a sting between quotes is provided, the program will read the entries separated by commas. Each entry will be analyzed individually in addition to whole system, that is always analyzed. *Example:* **`"moltype G1 G2, resname ALA, protein"`** analyses molecules G1 and G2 together, all residues ALA and all proteins. For more selection options see "Selection commands" section in the MDAnalysis manual

`-v` verbose, more text about what is happening, as well as some random dinosaurs

`-h` help menu with keywords descriptions

*Briefly, all major MD simulation engines' outputs are supported, including Amber, Gromacs, NAMD and LAMMPS.

4. OUTPUT FILES AND IMAGES

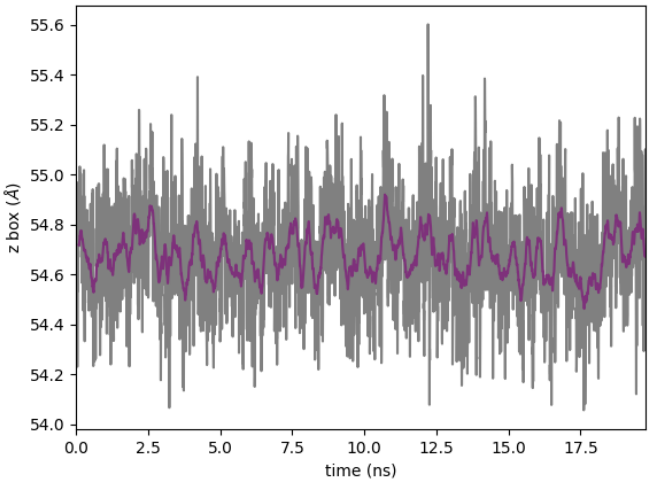
4.1 Text files

`dynden.py` generates a series of text files. `*.npy` files contain data in binary format. These contain results of calculations performed by *DynDen* and are loaded in subsequent executions of the code to speed up runtime. `*.dat` files contain processed data and are in text format.

- `bkp_box_dims.dat`: a file containing *z*-axis evolution, used to produce `fig_z_box.png`. Columns are time, min, max, dimensions.
- `bkp_result_XYZ.npy`: linear density of each frame for system component *XYZ*, used to produce `fig_density_XYZ_b_nf.png`.
- `bkp_correlation_all_traces_b_nf.dat`: a file containing the evolution of correlation of consecutive frames for each component (in columns), where *b* is the number of bins used, and *nf* is the number of time frames used for averaging. Data is used to produce `fig_correlation_all_traces_b_nf.png`.
- `dynden.log`: copy of all text displayed on screen, dinosaurs included.

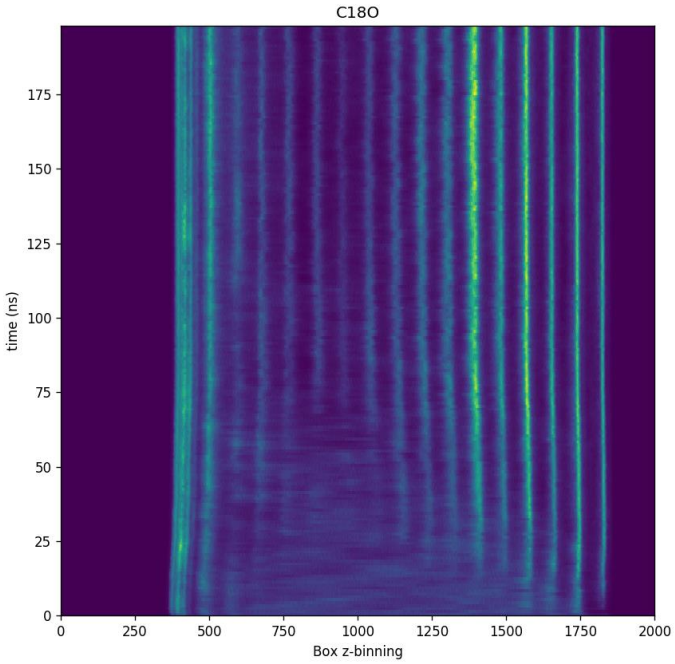
4.2 Images

- `fig_z_box.png`: time evolution of z-axis length. In grey box size at every frame, in palatinate an average calculated with a sliding window of a size of 1/100 of the total number of frames in the simulation.

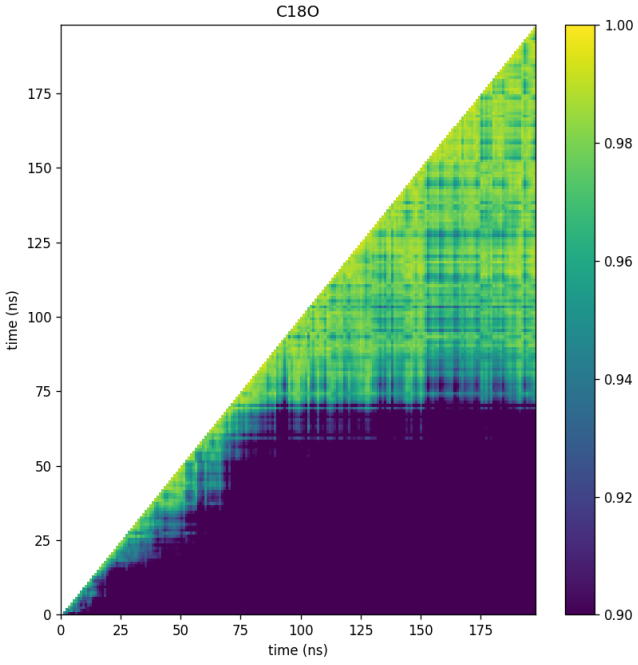


- `fig_density_XYZ_b_nf.png`: A surface plot showing the DTS (linear Density Time Series) for the component *XYZ* of the system or system itself. The colour represents the probability density at each moment in time (the lighter colour - the

higher the density). In the filename, *b* indicates the number of bins used, and *nf* is the number of time frames used for averaging.



- **`fig_PDC_XYZ_b_nf.png`:** All time frames *vs* all time frames correlation coefficient. Each coordinate represents the correlation between two given frames, on the diagonal we find consecutive frames (the values plotted on the `fig_correlation_all_traces_b_nf.png`). In the filename, *b* indicates the number of bins used, and *nf* is the number of time frames used for averaging.



- **`fig_correlation_all_traces_b_nf.png`**: Superimposition of all correlation coefficient trends of consecutive frames for every system component. In the filename, *b* indicates the number of bins used, and *nf* is the number of time frames used for averaging.