Discovery of new NSD2 non-histone substrates and design of a super-substrate

This repository provides python scripts for running and analysing molecular dynamics (MD) simulations of Protein-Lysine-Methyltransferases (PKMT) and their peptide substrate. The scripts are built based on the tools provided by OpenMM, MDTraj and contact map explorer.

With the scripts it is possible to:

- Produce molecular dynamics simulation trajectories using OpenMM on a ubuntu based GPU server.
    - Simulate the complex of a bound SAM molecule and PKMT with a complexed peptide (***Production_NSD2.py***)
- Analyse the of properties of the peptide and the interplay with the PKMT.
    - Analyse the number transition-state (TS)-like conformation between peptide and PKMT based on geometric S_N2-criteria (***Analysis_SN2-Transistion_States.py***).
    - Analyse the contacts between peptide and PKMT (***contact_map_explorer.py***).
- Parameter files (.xml) are given for: SAM and Zinc dummy (ZNB) atoms, gaff.xml can be found in this repo or downloaded here: https://github.com/choderalab/openmoltools/blob/master/openmoltools/parameters/gaff.xml

The scripts were written for the PKMT NSD2, peptides H3K36, ssK36 and SAM, but can be used for any other PKMT with other substrates. More comments and description of the variables, functions and outputs are provided within the individual scripts.

Installation

The MD simulation production and analysis scripts were designed for ubuntu based GPU servers using the cuda toolkit (v.11). The simulation engine is OpenMM (https://openmm.org/). For the analysis multiple tools were used e.g.,

- MDtraj (https://www.mdtraj.org/1.9.8.dev0/index.html)
- contact map explorer (https://contact-map.readthedocs.io/en/latest/examples/nb/contact_map.html)
- pandas (https://pandas.pydata.org/pandas-docs/stable/index.html)
- NumPy (https://numpy.org/)
- itertools (https://docs.python.org/3/library/itertools.html)

complete list can be found in every individual script in the first lines.
The easiest way to install the tools is with ***conda*** or ***miniconda***. Conda is a powerful package and environment management system. If you do not already have a highly customized Python environment, we recommend starting by installing miniconda.
Either in the full anaconda distribution (https://www.anaconda.com/) or the smaller footprint miniconda (https://docs.conda.io/en/latest/miniconda.html). Most of the tools are distributed through the conda-forge channel (https://conda-forge.org/); install it with *e.g.*:

```
conda install -c conda-forge contact_map
```

If you don't want to use conda, you can also use pip (not recommended) or do a developer install. See the individual installation documentations for details. Exact version for each tool can be found in the metadata of this repository.

Support

If you have suggestions or bug reports, please do not hesitate and contact Philipp Schnee or Albert Jeltsch (https://www.ibtb.uni-stuttgart.de/bc/team/).