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.

Python scripts

Python scripts in this repository:

- Analysis-SN2-Transition_States.py
- contact_map_explorer.py
- Production_NSD2.py

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 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).

The scripts were written for the PKMT NSD2, SAM and peptides H3K36 and ssK36, 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.

PDB files

PDB files provided in this repository can be used to reproduce MD simulation results described in the corresponding publication. PDB files provided in this repository are:

- NSD2_H3K36.pdb -> *Starting structure of complex of NSD2 with H3K36 peptide.*
- NSD2_H3K36_P30G_P43S.pdb -> *Starting structure of complex of NSD2 with H3K36 peptide with the mutations P30G and P43S, to represent the sequence as on the peptide array.*
- NSD2_H3K36_end_structure.pdb -> *End structure after 100 ns MD simulation of NSD2 with H3K36 peptide.*
- NSD2_ssK36.pdb -> *Starting structure of complex of NSD2 with sK36 peptide.*
- NSD2_ssK36_ P30G_P43S.pdb -> *Starting structure of complex of NSD2 with ssK36 peptide with the mutations P30G and P43S, to represent the sequence as on the peptide array.*
- NSD2_ssK36_end_structure.pdb -> *End structure after 100 ns MD simulation of NSD2 with ssK36 peptide.*
- NSD2_SAM.pdb -> *Starting structure of cofactor SAM.*
- NSD2_ZNB.pdb -> *Starting structure of zinc ion dummy model.*

The PDB files in this repo can directly be used as an input. Combine NSD2_H3K36.pdb or NSD2_ssK36.pdb with SAM.pdb and ZNB.pdb to generate your starting structure.

XML files

XML files contain parameters for the set up of MD simulations for molecules SAM and the zinc ion dummy. XML files provided in this repository are:

- SAM.xml
- ZNB.xml

Parameter files (.xml) are given for: SAM and Zinc dummy (ZNB) atoms, Gaff.xml can be downloaded here:
https://github.com/choderalab/openmoltools/blob/master/openmoltools/parameters/gaff.xml

Data sheet

Data sheet with all supplementary information and raw data obtained during MD simulation analysis. Data sheet provided in this repository is:

- MD SourceData.xlsx -> *this file contains all source data obtained from MD simulation conducted with the H3K36 and ssK36 peptide. This file also contains the data for the MD simulation with an independent equilibration before each replicate.*
- MD SourceData_P30G_P43S.xlsx -> *this file contains all source data obtained from MD simulation conducted with the H3K36 and ssK36 peptide containing the P30G and P43S mutation to represent the sequence of the peptide array.*

Movie

The movie “Movie_NSD2_ssK36.mp4” shows a snippet of a 100 ns MD simulation replicate of NSD2 complexed with the ssK36 peptide (green) and SAM (orange). The methyl group is coloured black and the target lysine pink. The zinc-ion dummy models are represented as slightly transparent cyan spheres.

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/).