Preferential interaction of DNMT3A subunits containing the R882H cancer mutation leads to dominant changes of flanking sequence effects

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

Python scripts

Python scripts in this repository:

- DNMT3A-Production.py
- DNMT3A-Contact_Maps.py
- DNMT3A-Contact_Maps_Analysis.py
- DNMT3A-Contact_Maps_Statistics.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 DNA double strand molecule and DNMT with a complexed SAM (DNMT3A-Production.py).

Analyse the contacts between peptide and PKMT (DNMT3A-Contact_Maps.py).

The scripts were written for DNMT3A WT and DNMT3A R882H, but can be used for any other DNMT. 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:

- DNMT3A-WT.pdb -> *Starting structure of complex of DNMT3A WT with DNA substrate*
- DNMT3A-MT.pdb -> *Starting structure of complex of DNMT3A R882H with DNA substrate*
- SMA.pdb -> *Starting structure of complex of one SAM molecule.*
- SMB.pdb -> *Starting structure of complex of the second SAM molecule.*

The PDB files in this repo can directly be used as an input.

XML files

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

- SMA.xml
- SMB.xml

gaff.xml can be found in this repo or downloaded here:
https://github.com/choderalab/openmoltools/blob/master/openmoltools/parameters/gaff.xml

Tools

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/).