Mechanistic basis of the increased methylation activity of the SETD2 protein lysine methyltransferase towards a designed super-substrate peptide

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, enspare 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 a peptide in a water box (***Production_Peptide_in_Solution.py***).
  - Simulate the association process of a peptide into the binding cleft of a PKMT (***Production_SETD2_sMD.py***)
  - Simulate the complex of a peptide in the binding cleft of a PKMT (***Production_SETD2-peptide-complex.py***)
- Analyse the of properties of the peptide and the interplay with the PKMT.
  - Clustering of the conformations from the peptide in solution simulations (***Enspara_Command.txt***, followed by ***Enspara_Clustering_Analysis.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 (***Analysis_Contact_Maps.py***, followed by ***Analysis_Contact_Maps_Analysis.py***).
  - Analyse the Unfolding of a peptide in the association process into the PKMT binding cleft (***Analysis_Peptide_Unfolding.py***). For a detailed analyses of the RMSD and End-to-End distance of the peptide see: ***Analysis_RMSD-C_N_Distance_Correlation.py***

The scripts were written for the PKMT SETD2 and the peptide substrates H3K36 and ssK36 but can be used for any other PKMT or peptide substrate. 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)
- enspara (https://enspara.readthedocs.io/en/latest/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)
- 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***. 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 conda.
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:

```
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/).