タンパク質の共溶媒分子動力学計算結果解析ソフトウェアの構築 マニュアル

株式会社 情報数理バイオ

2023/06/05

# ファイル一覧

* cosmdanalyzer/ : ソースコード
  + cosmdanalyzer.py : 実行用スクリプト
  + pyproject.toml : poetry用パッケージ管理ファイル
  + Dockerfile : docker-image作成用ファイル
  + src/ : ソースコード本体
  + data/ : 実行時に必要な静的データ類
    - setting.toml : 本プログラムの設定ファイルのデフォルト値
* sample\_data/ : サンプル用入力
  + single : 単一プローブサンプル
  + multi : 複数プローブサンプル
* doc : ドキュメント類(拡張子の違いはファイル形式のみで内容は同じ)
  + README.xxx : このドキュメント, 利用説明書

# インストール方法

## Dockerを利用した方法

Dockerが利用可能な環境が必要となる.  
Dockerfileのあるディレクトリで以下のコマンドを実行する.

docker image build ./ -t cosmdanalyzer

Dockerイメージがcosmdanalyzerの名前で作成される.

## poetryを利用した方法

python3.10以上とpoetryが利用可能な環境が必要となる.  
pyproject.tomlのあるディレクトリで以下のコマンドを実行する.

poetry install --no-dev

poetry run python ./cosmdanalyzer.py [オプション](#オプション)  
で実行可能になる.

## 直接依存環境を導入する方法

python3.10以上が利用可能な環境が必要となる.  
以下のコマンドで必要なpythonのパッケージをインストールする.

pip3 install rdkit griddataformats plotly tomli

# 使用方法

## サンプル入力を使用したコマンド例

### ローカル環境での実行

cosmdanalyzer.py のあるディレクトリで以下のコマンドを実行する.

単一プローブの場合

python cosmdanalyzer.py \  
 ../out/ \  
 ../sample\_input/single/ \  
 --output\_detail \  
 -v

複数プローブの場合

python cosmdanalyzer.py \  
 ../out/ \  
 ../sample\_input/multi/ \  
 --output\_detail \  
 -v

../out/ のディレクトリに結果が出力される.

### dockerを使う場合

cosmdanalyzer.py のあるディレクトリで以下のコマンドを実行する.

単一プローブの場合

docker container run --rm -v $(pwd)/../sample\_input/:/home/input/ \  
 -v $(pwd)/../out/:/home/out cosmdanalyzer \  
 /home/out/ \  
 /home/input/single/ \  
 --output\_detail \  
 -v

複数プローブの場合

docker container run --rm -v $(pwd)/../sample\_input/:/home/input/ \  
 -v $(pwd)/../out/:/home/out cosmdanalyzer \  
 /home/out/ \  
 /home/input/multi/ \  
 --output\_detail \  
 -v

dockerのオプションの後に本プログラムのオプションを続けると、本プログラムがdockerコンテナ内で実行される。

## オプション

usage: cosmdanalyzer.py [-h] [-s SETTING] [--fpocket\_info FPOCKET\_INFO]  
 [--fpocket\_pdb FPOCKET\_PDB] [--output\_detail]  
 out\_dir src\_pdb src\_open\_dx [src\_open\_dx ...]

* positional arguments:
  + out\_dir 出力ディレクトリ
  + src\_dir 入力ディレクトリ
* options:
  + -h, –help show this help message and exit
  + -s SETTING, –setting SETTING 設定ファイルのパス
  + –fpocket\_info FPOCKET\_INFO fpocketの出力\*\_info.txtファイルのパス
  + –fpocket\_pdb FPOCKET\_PDB fpocketの出力PDBファイルのパス
  + –output\_detail スコアの傾向を表す詳細情報を出力する
  + -v, –verbose 標準出力に詳細な処理情報を表示する

## 入力ディレクトリ

xxx\_nVH.dxファイルの存在するディレクトリを1つの系とする。 系のディレクトリは以下とする. 複数のPDBを入力とする場合

xxx\_nVH.dx  
fpocket\_out/  
system00/xxx\_position\_check2.pdb  
system01/xxx\_position\_check2.pdb  
...

単一のPDBを入力とする場合

xxx\_nVH.dx  
xxx\_position\_check2.pdb  
fpocket\_out/

* xxx\_nVH.dx : 共溶媒の占有率ファイル
* xxx\_position\_check2.pdb : トラジェクトリPDB
* fpocket\_out : fpocketの出力ディレクトリ

xxx\_nVH.dxが複数ある場合,複数プローブのスポットの合成計算を行う。  
ただし、xxx\_nVH.dxが存在する系のディレクトリの子ディレクトリにある xxx\_nVH.dxは無視される。

## 設定ファイル

設定ファイルはtoml形式で記述する。  
デフォルト設定ファイル

[clustering]  
# algorithm = "single\_linkage" or "dbscan" or "mean\_shift"  
algorithm = "dbscan"   
occupancy = 0.00001  
extend = 0.0  
spot\_marge\_rate = 0.5  
  
[clustering.single\_linkage]  
threshold = 3.0  
  
[clustering.dbscan]  
epsilon = 3.0  
min\_pts = 10  
  
[clustering.mean\_shift]  
bandwidth = 3.0  
  
[score]  
temperature = 300.0  
solvent\_radius = 1.4  
resolution = 256  
fpocket\_threshold = 0.0  
  
[score.weight]  
gfe = 1.0  
size = 1.0  
protrusion = 1.0  
convexity = 1.0  
compactness = 1.0  
hydrophobicity = 1.0  
charge\_density = 1.0  
flexibility = 1.0  
fpocket = 1.0

* clustering : クラスタリングの設定値
  + algorithm : スポットのクラスタリングアルゴリズム  
    “single-linkage” or “DBSCAN” or “mean-shift” から選択する。
  + occupancy : ボクセル選択時のしきい値  
    (プローブ占有確率/プローブ重原子数 ) >= occupancy のボクセルを選択
  + extend : スポットのボクセルを指定Å分拡大する
  + spot\_marge\_rate : [0.0, 1.0]複数プローブ入力時に指定割合以上重なっているスポットを同一スポットとする.
* score : スポットのスコア計算の設定値
  + temperature : 入力トラジェクトリ作成時の絶対温度(K)
  + solvent\_radius : 溶媒半径
  + resolution : 1原子の球表面を指定頂点数の多面体で近似する. 大きいほど精度が高く,該当部分の計算時間が比例して増加する.
  + weight : スコアの各項の重み
  + fpocket\_threshold : [0.0, 1.0] fpocket出力のポケットがスポットと指定割合以上重なっている場合スポットのスコアとする.

## 出力

系のディレクトリ名(outputの場合はその親ディレクトリ)をbasenameとしたとき以下のファイルが出力ディレクトリに生成される.

* all\_info.txt : すべてのプローブに対するスコア(フレーム数荷重平均)
* spot\_probe.toml : スポットとプローブの対応表
* basename/ : 各プローブ毎のディレクトリ
  + basename\_info.txt : スコアファイル
  + basename.pml : PyMol用入力
  + basename\_out.pdb : 入力PDBの最終フレームにスポットの立体配置を原子名APOLで追加したファイル
    - spots : スポット毎に対応するパッチ原子を記録したPDB
  + score\_detail.csv : 各スコアの傾向 (平均, 分散, 最小値, 最大値, 第一四分位数, 中央値, 第三四分位数) オプション –output\_detail を指定した場合のみ出力

### スコアファイル

出力例

Patch 1  
 score : 394.091  
 gfe : 1.337  
 size : 388.927  
 protrusion : 1.000  
 convexity : 0.022  
 compactness : 3.307  
 hydrophobicity : -0.930  
 charge\_density : -0.015  
 flexibility : 0.442  
 fpocket : n/a  
  
Patch 2  
...

パッチ毎にスコアの各項が出力される.

### PyMol用入力

出力されたbasename.pmlをPyMolで読み込むとスポットの立体構造が可視化される。

### score\_detail.csv

各列の情報は以下の通り

* spot : スポットの番号
* score\_type : 行のスコア値が表しているスコアの種類
* mean : 平均
* variance : 分散
* 0.0 : 最小値
* 0.25 : 第一四分位数
* 0.5 : 中央値
* 0.75 : 第三四分位数
* 1.0 : 最大値

出力例

spot,score\_type,mean,variance,0.0,0.25,0.50,0.75,1.0  
1,size,388.9274,19.73281,379.7122,385.784,388.1892,391.6664,398.7536  
1,protrusion,1,0,1,1,1,1,1  
1,convexity,0.02166458,5.456679e-06,0.01765554,0.02062114,0.02174732,0.02299604,0.0277241  
1,compactness,3.307329,0.07657145,2.908801,3.11562,3.170944,3.561596,3.787375  
1,charge\_density,-0.01541033,4.736121e-06,-0.01944439,-0.01679969,-0.01558867,-0.01391103,-0.01187199  
2,size,524.7099,17.29361,519.2607,521.8222,524.3902,526.5038,533.8469  
...

### スポットとプローブの対応ファイル

スポットに対応する複数のプローブを

スポット名 = プローブ名, プローブ名, …

の形式で記録する。

出力例

Patch 0 = A00  
Patch 1 = A01, A00  
...