Docking

November 5, 2022

1 Docking

Aim: Evaluate whether the 3d structure of the 2d-active compounds is active as well and if it poses in the binding pocket

[1]:

```
!pip install oddt
# And we need to install vina
!pip install vina
!pip install py3Dmol
!pip install MDAnalysis
!pip install nglview
!pip install rdkit
```

```
Collecting oddt
  Using cached oddt-0.7-py2.py3-none-any.whl
Requirement already satisfied: joblib>=0.9.4 in /opt/conda/lib/python3.9/site-
packages (from oddt) (1.1.0)
Collecting numpydoc
  Using cached numpydoc-1.5.0-py3-none-any.whl (52 kB)
Requirement already satisfied: pandas>=0.19.2 in /opt/conda/lib/python3.9/site-
packages (from oddt) (1.4.2)
Requirement already satisfied: numpy>=1.11 in /opt/conda/lib/python3.9/site-
packages (from oddt) (1.21.6)
Requirement already satisfied: six in /opt/conda/lib/python3.9/site-packages
(from oddt) (1.16.0)
Requirement already satisfied: scikit-learn>=0.18 in
/opt/conda/lib/python3.9/site-packages (from oddt) (1.1.1)
Requirement already satisfied: scipy>=0.17 in /opt/conda/lib/python3.9/site-
packages (from oddt) (1.7.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
/opt/conda/lib/python3.9/site-packages (from pandas>=0.19.2->oddt) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.9/site-
packages (from pandas>=0.19.2->oddt) (2022.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from scikit-learn>=0.18->oddt) (3.1.0)
Requirement already satisfied: Jinja2>=2.10 in /opt/conda/lib/python3.9/site-
packages (from numpydoc->oddt) (3.1.1)
Collecting sphinx>=4.2
```

```
  Using cached sphinx-5.3.0-py3-none-any.whl (3.2 MB)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.9/site-
packages (from Jinja2>=2.10->numpydoc->oddt) (2.1.1)
Collecting sphinxcontrib-htmlhelp>=2.0.0
  Using cached sphinxcontrib_htmlhelp-2.0.0-py2.py3-none-any.whl (100 kB)
Requirement already satisfied: requests>=2.5.0 in /opt/conda/lib/python3.9/site-
packages (from sphinx>=4.2->numpydoc->oddt) (2.27.1)
Requirement already satisfied: Pygments>=2.12 in /opt/conda/lib/python3.9/site-
packages (from sphinx>=4.2->numpydoc->oddt) (2.12.0)
Requirement already satisfied: packaging>=21.0 in /opt/conda/lib/python3.9/site-
packages (from sphinx>=4.2->numpydoc->oddt) (21.3)
Collecting docutils<0.20,>=0.14
  Using cached docutils-0.19-py3-none-any.whl (570 kB)
Collecting sphinxcontrib-jsmath
  Using cached sphinxcontrib_jsmath-1.0.1-py2.py3-none-any.whl (5.1 kB)
Collecting sphinxcontrib-devhelp
  Using cached sphinxcontrib_devhelp-1.0.2-py2.py3-none-any.whl (84 kB)
Collecting alabaster<0.8,>=0.7
  Using cached alabaster-0.7.12-py2.py3-none-any.whl (14 kB)
Collecting sphinxcontrib-qthelp
  Using cached sphinxcontrib_qthelp-1.0.3-py2.py3-none-any.whl (90 kB)
Collecting imagesize>=1.3
  Using cached imagesize-1.4.1-py2.py3-none-any.whl (8.8 kB)
Requirement already satisfied: importlib-metadata>=4.8 in
/opt/conda/lib/python3.9/site-packages (from sphinx>=4.2->numpydoc->oddt)
(4.11.3)
Collecting snowballstemmer>=2.0
  Using cached snowballstemmer-2.2.0-py2.py3-none-any.whl (93 kB)
Collecting sphinxcontrib-serializinghtml>=1.1.5
  Using cached sphinxcontrib_serializinghtml-1.1.5-py2.py3-none-any.whl (94 kB)
Collecting sphinxcontrib-applehelp
  Using cached sphinxcontrib_applehelp-1.0.2-py2.py3-none-any.whl (121 kB)
Requirement already satisfied: babel>=2.9 in /opt/conda/lib/python3.9/site-
packages (from sphinx>=4.2->numpydoc->oddt) (2.9.1)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.9/site-
packages (from importlib-metadata>=4.8->sphinx>=4.2->numpydoc->oddt) (3.8.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.9/site-packages (from
packaging>=21.0->sphinx>=4.2->numpydoc->oddt) (3.0.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.9/site-packages (from
requests>=2.5.0->sphinx>=4.2->numpydoc->oddt) (1.26.9)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.9/site-packages (from
requests>=2.5.0->sphinx>=4.2->numpydoc->oddt) (2022.6.15)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from
requests>=2.5.0->sphinx>=4.2->numpydoc->oddt) (2.0.12)
```

```
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.9/site-
packages (from requests>=2.5.0->sphinx>=4.2->numpydoc->oddt) (3.3)
Installing collected packages: snowballstemmer, alabaster, sphinxcontrib-
serializinghtml, sphinxcontrib-qthelp, sphinxcontrib-jsmath, sphinxcontrib-
htmlhelp, sphinxcontrib-devhelp, sphinxcontrib-applehelp, imagesize, docutils,
sphinx, numpydoc, oddt
Successfully installed alabaster-0.7.12 docutils-0.19 imagesize-1.4.1
numpydoc-1.5.0 oddt-0.7 snowballstemmer-2.2.0 sphinx-5.3.0 sphinxcontrib-
applehelp-1.0.2 sphinxcontrib-devhelp-1.0.2 sphinxcontrib-htmlhelp-2.0.0
sphinxcontrib-jsmath-1.0.1 sphinxcontrib-qthelp-1.0.3 sphinxcontrib-
serializinghtml-1.1.5
Collecting vina
  Using cached
vina-1.2.3-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.1 MB)
Requirement already satisfied: numpy>=1.18 in /opt/conda/lib/python3.9/site-
packages (from vina) (1.21.6)
Installing collected packages: vina
Successfully installed vina-1.2.3
Collecting py3Dmol
  Using cached py3Dmol-1.8.1-py2.py3-none-any.whl (6.5 kB)
Installing collected packages: py3Dmol
Successfully installed py3Dmol-1.8.1
Collecting MDAnalysis
  Using cached
MDAnalysis-2.3.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.6
MB)
Collecting biopython>=1.71
  Using cached
biopython-1.79-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.7 MB)
Requirement already satisfied: packaging in /opt/conda/lib/python3.9/site-
packages (from MDAnalysis) (21.3)
Requirement already satisfied: joblib>=0.12 in /opt/conda/lib/python3.9/site-
packages (from MDAnalysis) (1.1.0)
Requirement already satisfied: numpy>=1.20.0 in /opt/conda/lib/python3.9/site-
packages (from MDAnalysis) (1.21.6)
Collecting GridDataFormats>=0.4.0
  Using cached GridDataFormats-1.0.1-py3-none-any.whl (2.1 MB)
Requirement already satisfied: networkx>=2.0 in /opt/conda/lib/python3.9/site-
packages (from MDAnalysis) (2.8.5)
Collecting gsd>=1.9.3
  Using cached
gsd-2.6.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (396 kB)
Requirement already satisfied: threadpoolctl in /opt/conda/lib/python3.9/site-
packages (from MDAnalysis) (3.1.0)
Requirement already satisfied: scipy>=1.5.0 in /opt/conda/lib/python3.9/site-
packages (from MDAnalysis) (1.7.3)
Requirement already satisfied: tqdm>=4.43.0 in /opt/conda/lib/python3.9/site-
packages (from MDAnalysis) (4.64.0)
```

```
Collecting fasteners
  Using cached fasteners-0.18-py3-none-any.whl (18 kB)
Requirement already satisfied: matplotlib>=1.5.1 in
/opt/conda/lib/python3.9/site-packages (from MDAnalysis) (3.5.2)
Collecting mmtf-python>=1.0.0
  Using cached mmtf_python-1.1.3-py2.py3-none-any.whl (25 kB)
Collecting mrcfile
  Using cached mrcfile-1.4.3-py2.py3-none-any.whl (43 kB)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.9/site-packages (from matplotlib>=1.5.1->MDAnalysis)
(2.8.2)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.9/site-
packages (from matplotlib>=1.5.1->MDAnalysis) (9.1.1)
Requirement already satisfied: pyparsing>=2.2.1 in
/opt/conda/lib/python3.9/site-packages (from matplotlib>=1.5.1->MDAnalysis)
(3.0.8)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.9/site-
packages (from matplotlib>=1.5.1->MDAnalysis) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.9/site-packages (from matplotlib>=1.5.1->MDAnalysis)
(4.34.4)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.9/site-packages (from matplotlib>=1.5.1->MDAnalysis)
(1.4.4)
Requirement already satisfied: msgpack>=1.0.0 in /opt/conda/lib/python3.9/site-
packages (from mmtf-python>=1.0.0->MDAnalysis) (1.0.4)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.9/site-
packages (from python-dateutil>=2.7->matplotlib>=1.5.1->MDAnalysis) (1.16.0)
Installing collected packages: mrcfile, mmtf-python, gsd, fasteners, biopython,
GridDataFormats, MDAnalysis
Successfully installed GridDataFormats-1.0.1 MDAnalysis-2.3.0 biopython-1.79
fasteners-0.18 gsd-2.6.0 mmtf-python-1.1.3 mrcfile-1.4.3
Collecting nglview
  Using cached nglview-3.0.3-py3-none-any.whl
Requirement already satisfied: numpy in /opt/conda/lib/python3.9/site-packages
(from nglview) (1.21.6)
Requirement already satisfied: jupyterlab-widgets in
/opt/conda/lib/python3.9/site-packages (from nglview) (1.1.1)
Requirement already satisfied: ipywidgets>=7 in /opt/conda/lib/python3.9/site-
packages (from nglview) (7.7.0)
Requirement already satisfied: widgetsnbextension~=3.6.0 in
/opt/conda/lib/python3.9/site-packages (from ipywidgets>=7->nglview) (3.6.0)
Requirement already satisfied: ipykernel>=4.5.1 in
/opt/conda/lib/python3.9/site-packages (from ipywidgets>=7->nglview) (6.13.0)
Requirement already satisfied: ipython>=4.0.0 in /opt/conda/lib/python3.9/site-
packages (from ipywidgets>=7->nglview) (8.3.0)
Requirement already satisfied: traitlets>=4.3.1 in
/opt/conda/lib/python3.9/site-packages (from ipywidgets>=7->nglview) (5.1.1)
```

```
Requirement already satisfied: ipython-genutils~=0.2.0 in
/opt/conda/lib/python3.9/site-packages (from ipywidgets>=7->nglview) (0.2.0)
Requirement already satisfied: nbformat>=4.2.0 in /opt/conda/lib/python3.9/site-
packages (from ipywidgets>=7->nglview) (5.3.0)
Requirement already satisfied: debugpy>=1.0 in /opt/conda/lib/python3.9/site-
packages (from ipykernel>=4.5.1->ipywidgets>=7->nglview) (1.6.0)
Requirement already satisfied: matplotlib-inline>=0.1 in
/opt/conda/lib/python3.9/site-packages (from
ipykernel>=4.5.1->ipywidgets>=7->nglview) (0.1.3)
Requirement already satisfied: packaging in /opt/conda/lib/python3.9/site-
packages (from ipykernel>=4.5.1->ipywidgets>=7->nglview) (21.3)
Requirement already satisfied: tornado>=6.1 in /opt/conda/lib/python3.9/site-
packages (from ipykernel>=4.5.1->ipywidgets>=7->nglview) (6.1)
Requirement already satisfied: nest-asyncio in /opt/conda/lib/python3.9/site-
packages (from ipykernel>=4.5.1->ipywidgets>=7->nglview) (1.5.5)
Requirement already satisfied: psutil in /opt/conda/lib/python3.9/site-packages
(from ipykernel>=4.5.1->ipywidgets>=7->nglview) (5.9.0)
Requirement already satisfied: jupyter-client>=6.1.12 in
/opt/conda/lib/python3.9/site-packages (from
ipykernel>=4.5.1->ipywidgets>=7->nglview) (7.3.0)
Requirement already satisfied: backcall in /opt/conda/lib/python3.9/site-
packages (from ipython>=4.0.0->ipywidgets>=7->nglview) (0.2.0)
Requirement already satisfied: stack-data in /opt/conda/lib/python3.9/site-
packages (from ipython>=4.0.0->ipywidgets>=7->nglview) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in
/opt/conda/lib/python3.9/site-packages (from
ipython>=4.0.0->ipywidgets>=7->nglview) (62.1.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from
ipython>=4.0.0->ipywidgets>=7->nglview) (3.0.29)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.9/site-
packages (from ipython>=4.0.0->ipywidgets>=7->nglview) (0.18.1)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.9/site-
packages (from ipython>=4.0.0->ipywidgets>=7->nglview) (4.8.0)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.9/site-
packages (from ipython>=4.0.0->ipywidgets>=7->nglview) (0.7.5)
Requirement already satisfied: pygments>=2.4.0 in /opt/conda/lib/python3.9/site-
packages (from ipython>=4.0.0->ipywidgets>=7->nglview) (2.12.0)
Requirement already satisfied: decorator in /opt/conda/lib/python3.9/site-
packages (from ipython>=4.0.0->ipywidgets>=7->nglview) (5.1.1)
Requirement already satisfied: jupyter-core in /opt/conda/lib/python3.9/site-
packages (from nbformat>=4.2.0->ipywidgets>=7->nglview) (4.9.2)
Requirement already satisfied: fastjsonschema in /opt/conda/lib/python3.9/site-
packages (from nbformat>=4.2.0->ipywidgets>=7->nglview) (2.15.3)
Requirement already satisfied: jsonschema>=2.6 in /opt/conda/lib/python3.9/site-
packages (from nbformat>=4.2.0->ipywidgets>=7->nglview) (4.4.0)
Requirement already satisfied: notebook>=4.4.1 in /opt/conda/lib/python3.9/site-
packages (from widgetsnbextension~=3.6.0->ipywidgets>=7->nglview) (6.4.11)
```

```
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/opt/conda/lib/python3.9/site-packages (from
jedi>=0.16->ipython>=4.0.0->ipywidgets>=7->nglview) (0.8.3)
Requirement already satisfied: attrs>=17.4.0 in /opt/conda/lib/python3.9/site-
packages (from jsonschema>=2.6->nbformat>=4.2.0->ipywidgets>=7->nglview)
(21.4.0)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
/opt/conda/lib/python3.9/site-packages (from
jsonschema>=2.6->nbformat>=4.2.0->ipywidgets>=7->nglview) (0.18.1)
Requirement already satisfied: entrypoints in /opt/conda/lib/python3.9/site-
packages (from jupyter-client>=6.1.12->ipykernel>=4.5.1->ipywidgets>=7->nglview)
(0.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/lib/python3.9/site-packages (from jupyter-
client>=6.1.12->ipykernel>=4.5.1->ipywidgets>=7->nglview) (2.8.2)
Requirement already satisfied: pyzmq>=22.3 in /opt/conda/lib/python3.9/site-
packages (from jupyter-client>=6.1.12->ipykernel>=4.5.1->ipywidgets>=7->nglview)
(22.3.0)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.9/site-packages
(from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->nglview)
(3.1.1)
Requirement already satisfied: Send2Trash>=1.8.0 in
/opt/conda/lib/python3.9/site-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->nglview) (1.8.0)
Requirement already satisfied: prometheus-client in
/opt/conda/lib/python3.9/site-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->nglview) (0.7.1)
Requirement already satisfied: terminado>=0.8.3 in
/opt/conda/lib/python3.9/site-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->nglview) (0.13.3)
Requirement already satisfied: argon2-cffi in /opt/conda/lib/python3.9/site-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->nglview) (21.3.0)
Requirement already satisfied: nbconvert>=5 in /opt/conda/lib/python3.9/site-
packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->nglview) (6.5.0)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.9/site-
packages (from pexpect>4.3->ipython>=4.0.0->ipywidgets>=7->nglview) (0.7.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.9/site-packages
(from prompt-
toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=4.0.0->ipywidgets>=7->nglview)
(0.2.5)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.9/site-packages (from
packaging->ipykernel>=4.5.1->ipywidgets>=7->nglview) (3.0.8)
Requirement already satisfied: executing in /opt/conda/lib/python3.9/site-
packages (from stack-data->ipython>=4.0.0->ipywidgets>=7->nglview) (0.8.3)
Requirement already satisfied: pure-eval in /opt/conda/lib/python3.9/site-
```

```
packages (from stack-data->ipython>=4.0.0->ipywidgets>=7->nglview) (0.2.2)
Requirement already satisfied: asttokens in /opt/conda/lib/python3.9/site-
packages (from stack-data->ipython>=4.0.0->ipywidgets>=7->nglview) (2.0.5)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.9/site-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets>=7->nglview) (2.1.1)
Requirement already satisfied: nbclient>=0.5.0 in /opt/conda/lib/python3.9/site-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets>=7->nglview) (0.6.3)
Requirement already satisfied: jupyterlab-pygments in
/opt/conda/lib/python3.9/site-packages (from nbconvert>=5->notebook>=4.4.1->widg
etsnbextension~=3.6.0->ipywidgets>=7->nglview) (0.2.2)
Requirement already satisfied: bleach in /opt/conda/lib/python3.9/site-packages
(from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->n
glview) (5.0.0)
Requirement already satisfied: pandocfilters>=1.4.1 in
/opt/conda/lib/python3.9/site-packages (from nbconvert>=5->notebook>=4.4.1->widg
etsnbextension~=3.6.0->ipywidgets>=7->nglview) (1.5.0)
Requirement already satisfied: tinycss2 in /opt/conda/lib/python3.9/site-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets>=7->nglview) (1.1.1)
Requirement already satisfied: mistune<2,>=0.8.1 in
/opt/conda/lib/python3.9/site-packages (from nbconvert>=5->notebook>=4.4.1->widg
etsnbextension~=3.6.0->ipywidgets>=7->nglview) (0.8.4)
Requirement already satisfied: beautifulsoup4 in /opt/conda/lib/python3.9/site-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets>=7->nglview) (4.11.1)
Requirement already satisfied: defusedxml in /opt/conda/lib/python3.9/site-
packages (from nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidg
ets>=7->nglview) (0.7.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.9/site-
packages (from python-dateutil>=2.8.2->jupyter-
client>=6.1.12->ipykernel>=4.5.1->ipywidgets>=7->nglview) (1.16.0)
Requirement already satisfied: argon2-cffi-bindings in
/opt/conda/lib/python3.9/site-packages (from
argon2-cffi->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets>=7->nglview)
(21.2.0)
Requirement already satisfied: cffi>=1.0.1 in /opt/conda/lib/python3.9/site-
packages (from argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->widgetsnbexte
nsion~=3.6.0->ipywidgets>=7->nglview) (1.15.0)
Requirement already satisfied: soupsieve>1.2 in /opt/conda/lib/python3.9/site-
packages (from beautifulsoup4->nbconvert>=5->notebook>=4.4.1->widgetsnbextension
~=3.6.0->ipywidgets>=7->nglview) (2.3.1)
Requirement already satisfied: webencodings in /opt/conda/lib/python3.9/site-
packages (from bleach->nbconvert>=5->notebook>=4.4.1->widgetsnbextension~=3.6.0-
>ipywidgets>=7->nglview) (0.5.1)
Requirement already satisfied: pycparser in /opt/conda/lib/python3.9/site-
packages (from cffi>=1.0.1->argon2-cffi-bindings->argon2-cffi->notebook>=4.4.1->
widgetsnbextension~=3.6.0->ipywidgets>=7->nglview) (2.21)
Installing collected packages: nglview
Successfully installed nglview-3.0.3
Collecting rdkit
  Using cached
rdkit-2022.9.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (29.7
MB)
Requirement already satisfied: Pillow in /opt/conda/lib/python3.9/site-packages
(from rdkit) (9.1.1)
Requirement already satisfied: numpy in /opt/conda/lib/python3.9/site-packages
(from rdkit) (1.21.6)
Installing collected packages: rdkit
Successfully installed rdkit-2022.9.1
```

[2]:

```
#from pymol import cmd
import py3Dmol

import pandas as pd
import glob
import sys

from vina import Vina
#import pybel

from rdkit import Chem
from rdkit.Chem import AllChem, Draw

#from meeko import MoleculePreparation
#from meeko import obutils

import MDAnalysis as mda
from MDAnalysis.coordinates import PDB

#import prolif
#from prolif.plotting.network import LigNetwork

import nglview
#from scripts import viewer

import sys, os, shutil
#sys.path.insert(1, '/project/jhllei001/JHL_data/Jupyter_Dock/utilities')
# Note I had to comment out pymol, openmm and pdbfixer, hope it doesn't break later
#from utils import fix_protein, getbox, generate_ledock_file, pdbqt_to_sdf, dok_to_sdf

import warnings
```

```
warnings.filterwarnings("ignore")
%config Completer.use_jedi = False
```

[3]:
```
try:
    os.mkdir('Docking')
except:
    print("Directory already exists")
os.chdir('Docking')
```

```
Directory already exists
```

[4]:
```
PDBCODE = '2QBS' # As in lab 1, change this to your PDB accession code.
LIGAND_CODE = '024' # and add the ligand code
```

[5]:
```
active_de_novo = pd.read_csv("../Lab/data/Active_generated_molecules.csv")
test_smiles = active_de_novo['SMILES'].values
active_de_novo
```

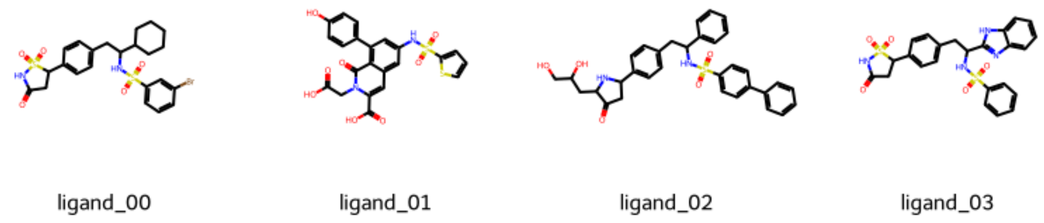
[5]:
```
   Unnamed: 0.1  Unnamed: 0  \
0            42         107
1            85         211
2           102         246
3           360         795

                                              SMILES  ro5_fulfilled  \
0  O=C1CC(c2ccc(CC(NS(=O)(=O)c3cccc(Br)c3)C3CCCCC...           True
1  O=C(O)Cn1c(C(=O)O)cc2cc(NS(=O)(=O)c3cccs3)cc(-...           True
2  O=C1CC(c2ccc(CC(NS(=O)(=O)c3ccc(-c4ccccc4)cc3)...           True
3  O=C1CC(c2ccc(CC(NS(=O)(=O)c3ccccc3)c3nc4ccccc4...           True

                                           ROMol  \
0  <rdkit.Chem.rdchem.Mol object at 0x7efc9c753dc0>
1  <rdkit.Chem.rdchem.Mol object at 0x7efc9c753e20>
2  <rdkit.Chem.rdchem.Mol object at 0x7efc9c753e80>
3  <rdkit.Chem.rdchem.Mol object at 0x7efc9c753ee0>

                                                  fp  activity
0  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...  0.600549
1  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...  0.818777
2  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...  0.906370
3  [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0...  0.858107
```

[6]:
```
test_smiles
```

```
[6]: array(['O=C1CC(c2ccc(CC(NS(=O)(=O)c3cccc(Br)c3)C3CCCCC3)cc2)S(=O)(=O)N1',
            'O=C(O)Cn1c(C(=O)O)cc2cc(NS(=O)(=O)c3cccs3)cc(-c3ccc(O)cc3)c2c1=O',
            'O=C1CC(c2ccc(CC(NS(=O)(=O)c3ccc(-c4ccccc4)cc3)c3ccccc3)cc2)NC1CC(O)CO',
            'O=C1CC(c2ccc(CC(NS(=O)(=O)c3ccccc3)c3nc4ccccc4[nH]3)cc2)S(=O)(=O)N1'],
           dtype=object)
```

```
[9]: i = -1

     IDs = []
     for i in range(0,len(test_smiles)):
         base = 'ligand_{:02d}'.format(i)
         IDs.append(base)

     df = pd.DataFrame({'ID':IDs, 'SMILES':test_smiles})

     df['rdkit_mol'] = [Chem.MolFromSmiles(s) for s in df['SMILES']]

     # Generate 2D images of the molecules
     img=Draw.MolsToGridImage(df['rdkit_mol'].tolist(),
                             molsPerRow=4,
                             subImgSize=(200,200),
                             legends=df['ID'].tolist())

     for i in range(len(df)):
         row = df.iloc[i]
         mol = row['rdkit_mol']
         name = row['ID'] + '.mol'
         Chem.MolToMolFile(mol,name)

     img
```

[9]:



```
[10]: view = py3Dmol.view()
      view.removeAllModels()
      view.setViewStyle({'style':'outline','color':'black','width':0.1})

      view.addModel(open('{}.pdb'.format(PDBCODE),'r').read(),format='pdb')
```

```
Prot=view.getModel()
Prot.setStyle({'cartoon':{'arrows':True, 'tubes':True, 'style':'oval', 'color':
 ↪'white'}})
view.addSurface(py3Dmol.VDW,{'opacity':0.6,'color':'white'})


view.addModel(open('ligand_00.mol','r').read(),format='mol2')
ref_m = view.getModel()
ref_m.setStyle({},{'stick':{'colorscheme':'greenCarbon','radius':0.2}})

ligtmp = 'ligand-{}.pdb'.format(LIGAND_CODE)
view.addModelsAsFrames(open(ligtmp,'r').read(),format='pdb')
ref_m = view.getModel()
ref_m.setStyle({},{'stick':{'colorscheme':'magentaCarbon','radius':0.2}})

view.zoomTo()
view.show()
```

[7]:

```
from math import isnan, isinf
def PDBQTAtomLines(mol, donors, acceptors):
    """Create a list with PDBQT atom lines for each atom in molecule. Donors
    and acceptors are given as a list of atom indices.
    """

    atom_lines = [line.replace('HETATM', 'ATOM  ')
                  for line in Chem.MolToPDBBlock(mol).split('\n')
                  if line.startswith('HETATM') or line.startswith('ATOM')]

    pdbqt_lines = []
    for idx, atom in enumerate(mol.GetAtoms()):
        pdbqt_line = atom_lines[idx][:56]

        pdbqt_line += '0.00  0.00    '  # append empty vdW and ele
        # Get charge
        charge = 0.
        fields = ['_MMFF94Charge', '_GasteigerCharge', '_TriposPartialCharge']
        for f in fields:
            if atom.HasProp(f):
                charge = atom.GetDoubleProp(f)
                break
        # FIXME: this should not happen, blame RDKit
        if isnan(charge) or isinf(charge):
            charge = 0.
        pdbqt_line += ('%.3f' % charge).rjust(6)

        # Get atom type
        pdbqt_line += ' '
        atomicnum = atom.GetAtomicNum()
        if atomicnum == 6 and atom.GetIsAromatic():
            pdbqt_line += 'A'
        elif atomicnum == 7 and idx in acceptors:
            pdbqt_line += 'NA'
        elif atomicnum == 8 and idx in acceptors:
            pdbqt_line += 'OA'
        elif atomicnum == 1 and atom.GetNeighbors()[0].GetIdx() in donors:
            pdbqt_line += 'HD'
        else:
            pdbqt_line += atom.GetSymbol()
        pdbqt_lines.append(pdbqt_line)
    return pdbqt_lines

def MolToPDBQTBlock(mol, flexible=True, addHs=False, computeCharges=False):
    """Write RDKit Molecule to a PDBQT block

    Parameters
    ----------
        mol: rdkit.Chem.rdchem.Mol
            Molecule with a protein ligand complex
        flexible: bool (default=True)
            Should the molecule encode torsions. Ligands should be flexible,
            proteins in turn can be rigid.
        addHs: bool (default=False)
            The PDBQT format requires at least polar Hs on donors. By default Hs
            are added.
        computeCharges: bool (default=False)
            Should the partial charges be automatically computed. If the Hs are
            added the charges must and will be recomputed. If there are no
            partial charge information, they are set to 0.0.

    Returns
    -------
        block: str
            String wit PDBQT encoded molecule
    """
    # make a copy of molecule
    mol = Chem.Mol(mol)

    # if flexible molecule contains multiple fragments write them separately
    if flexible and len(Chem.GetMolFrags(mol)) > 1:
        return ''.join(MolToPDBQTBlock(frag, flexible=flexible, addHs=addHs,
                                       computeCharges=computeCharges)
                       for frag in Chem.GetMolFrags(mol, asMols=True))

    # Identify donors and acceptors for atom typing
    # Acceptors
    patt = Chem.MolFromSmarts('[$([O;H1;v2]),'
                              '$([O;H0;v2;!$(O=N-*),'
                              '$([O;-;!$(*-N=O)]),'
                              '$([o;+0])]),'
                              '$([n;+0;!X3;!$([n;H1](cc)cc),'
                              '$([$([N;H0]#[C&v4])]),'
                              '$([N&v3;H0;$(Nc)])]),'
                              '$([F;$(F-[#6]);!$(FC[F,Cl,Br,I])])]')
    acceptors = list(map(lambda x: x[0],
                         mol.GetSubstructMatches(patt, maxMatches=mol.
↪GetNumAtoms())))
    # Donors
    patt = Chem.MolFromSmarts('[$([N&!H0&v3,N&!
↪H0&+1&v4,n&H1&+0,$([$([Nv3](-C)(-C)-C)]),'
                              '$([$(n[n;H1]),'
                              '$(nc[n;H1])])]),'
                              # Guanidine can be tautormeic - e.g. Arginine
                              '$([NX3,NX2]([!O,!S])!@C(!@[NX3,NX2]([!O,!S]))!
↪@[NX3,NX2]([!O,!S])),'
                              '$([O,S;H1;+0])]')
    donors = list(map(lambda x: x[0],
                      mol.GetSubstructMatches(patt, maxMatches=mol.
↪GetNumAtoms())))
    if addHs:
        mol = Chem.AddHs(mol, addCoords=True, onlyOnAtoms=donors, )
    if addHs or computeCharges:
        AllChem.ComputeGasteigerCharges(mol)

    atom_lines = PDBQTAtomLines(mol, donors, acceptors)
    assert len(atom_lines) == mol.GetNumAtoms()

    pdbqt_lines = []

    # vina scores
    if (mol.HasProp('vina_affinity') and mol.HasProp('vina_rmsd_lb') and
            mol.HasProp('vina_rmsd_lb')):
        pdbqt_lines.append('REMARK VINA RESULT:  ' +
                           ('%.1f' % float(mol.GetProp('vina_affinity'))).
↪rjust(8) +
                           ('%.3f' % float(mol.GetProp('vina_rmsd_lb'))).
↪rjust(11) +
                           ('%.3f' % float(mol.GetProp('vina_rmsd_ub'))).
↪rjust(11))

    pdbqt_lines.append('REMARK  Name = ' +
                       (mol.GetProp('_Name') if mol.HasProp('_Name') else ''))
    if flexible:
        # Find rotatable bonds
        rot_bond = Chem.MolFromSmarts('[!$(*#*)&!D1&!$(C(F)(F)F)&'
                                      '!$(C(Cl)(Cl)Cl)&'
                                      '!$(C(Br)(Br)Br)&'
                                      '!$(C([CH3])([CH3])[CH3])&'
                                      '!$([CD3](=[N,O,S])-!@[#7,O,S!D1])&'
                                      '!$([#7,O,S!D1]-!@[CD3]=[N,O,S])&'
                                      '!$([CD3](=[N+])-!@[#7!D1])&'
                                      '!$([#7!D1]-!@[CD3]=[N+])]-!@[!$(*#*)&'
                                      '!D1&!$(C(F)(F)F)&'
                                      '!$(C(Cl)(Cl)Cl)&'
                                      '!$(C(Br)(Br)Br)&'
                                      '!$(C([CH3])([CH3])[CH3])]')
        bond_atoms = list(mol.GetSubstructMatches(rot_bond))
        num_torsions = len(bond_atoms)

        # Active torsions header
        pdbqt_lines.append('REMARK  %i active torsions:' % num_torsions)
        pdbqt_lines.append('REMARK  status: (\'A\' for Active; \'I\' for␣
↪Inactive)')
        for i, (a1, a2) in enumerate(bond_atoms):
            pdbqt_lines.append('REMARK%5.0i  A    between atoms: _%i  and  _%i'
                               % (i + 1, a1 + 1, a2 + 1))

        # Fragment molecule on bonds to ge rigid fragments
        bond_ids = [mol.GetBondBetweenAtoms(a1, a2).GetIdx()
                    for a1, a2 in bond_atoms]
        if bond_ids:
            mol_rigid_frags = Chem.FragmentOnBonds(mol, bond_ids,␣
↪addDummies=False)
        else:
            mol_rigid_frags = mol
        frags = list(Chem.GetMolFrags(mol_rigid_frags))

        def weigh_frags(frag):
            """sort by the fragment size and the number of bonds (secondary)"""
            num_bonds = 0
            # bond_weight = 0
            for a1, a2 in bond_atoms:
                if a1 in frag or a2 in frag:
                    num_bonds += 1
                    # for frag2 in frags:
                    #     if a1 in frag2 or a2 in frag2:
                    #         bond_weight += len(frag2)
            # changed signs are fixing mixed sorting type (ascending/descending)
            return -len(frag), -num_bonds,  # bond_weight
        frags = sorted(frags, key=weigh_frags)

        # Start writting the lines with ROOT
        pdbqt_lines.append('ROOT')
        frag = frags.pop(0)
        for idx in frag:
            pdbqt_lines.append(atom_lines[idx])
        pdbqt_lines.append('ENDROOT')

        # Now build the tree of torsions usign DFS algorithm. Keep track of last
        # route with following variables to move down the tree and close␣
↪branches
        branch_queue = []
        current_root = frag
        old_roots = [frag]

        visited_frags = []
        visited_bonds = []
        while len(frags) > len(visited_frags):
            end_branch = True
            for frag_num, frag in enumerate(frags):
                for bond_num, (a1, a2) in enumerate(bond_atoms):
                    if (frag_num not in visited_frags and
                        bond_num not in visited_bonds and
                        (a1 in current_root and a2 in frag or
                         a2 in current_root and a1 in frag)):
                        # direction of bonds is important
                        if a1 in current_root:
                            bond_dir = '%i %i' % (a1 + 1, a2 + 1)
                        else:
                            bond_dir = '%i %i' % (a2 + 1, a1 + 1)
                        pdbqt_lines.append('BRANCH %s' % bond_dir)
                        for idx in frag:
                            pdbqt_lines.append(atom_lines[idx])
                        branch_queue.append('ENDBRANCH %s' % bond_dir)

                        # Overwrite current root and stash previous one in queue
                        old_roots.append(current_root)
                        current_root = frag

                        # remove used elements from stack
                        visited_frags.append(frag_num)
                        visited_bonds.append(bond_num)

                        # mark that we dont want to end branch yet
```

```python
                        end_branch = False
                        break
                    else:
                        continue
                    break  # break the outer loop as well

            if end_branch:
                pdbqt_lines.append(branch_queue.pop())
                if old_roots:
                    current_root = old_roots.pop()
        # close opened branches if any is open
        while len(branch_queue):
            pdbqt_lines.append(branch_queue.pop())
        pdbqt_lines.append('TORSDOF %i' % num_torsions)
    else:
        pdbqt_lines.extend(atom_lines)

    return '\n'.join(pdbqt_lines)

def MolFromPDBQTBlock(filename, sanitize=True, removeHs=True):
    """Read PDBQT block to a RDKit Molecule
    Parameters
    ----------
        block: string
            Residue name which explicitly pint to a ligand(s).
        sanitize: bool (default=True)
            Should the sanitization be performed
        removeHs: bool (default=True)
            Should hydrogens be removed when reading molecule.
    Returns
    -------
        mol: rdkit.Chem.rdchem.Mol
            Molecule read from PDBQT
    """
    pdb_lines = []
    name = ''
    data = {}
    with open(filename) as file:
        lines = file.readlines()
        block = [line.rstrip() for line in lines]

    for line in block: #.split('\n'):
        # Get all know data from REMARK section
        if line[:12] == 'REMARK  Name':
            name = line[15:].strip()
        elif line[:18] == 'REMARK VINA RESULT':
            tmp = line[19:].split()
            data['vina_affinity'] = tmp[0]
            data['vina_rmsd_lb'] = tmp[1]
            data['vina_rmsd_ub'] = tmp[2]

        # no more data to collect
        if line[:4] != 'ATOM':
            continue

        pdb_line = line[:56]
        pdb_line += '1.00  0.00           '

        # Do proper atom type lookup
        atom_type = line[71:].split()[1]
        if atom_type == 'A':
            atom_type = 'C'
        elif atom_type[:1] == 'O':
            atom_type = 'O'
        elif atom_type[:1] == 'H':
            atom_type = 'H'
            if removeHs:
                continue
        elif atom_type == 'NA':
            atom_type = 'N'

        pdb_lines.append(pdb_line + atom_type)
    mol = Chem.MolFromPDBBlock('\n'.join(pdb_lines), sanitize=False)
    if sanitize:
        Chem.SanitizeMol(mol)
    else:
        Chem.GetSSSR(mol)
    # reorder atoms using serial
    new_order = sorted(range(mol.GetNumAtoms()),
                       key=lambda i: (mol.GetAtomWithIdx(i)
                                      .GetPDBResidueInfo()
                                      .GetSerialNumber()))
    mol = Chem.RenumberAtoms(mol, new_order)

    # properties must be set on final copy of Mol, RenumberAtoms purges data
    mol.SetProp('_Name', name)
    for k, v in data.items():
        mol.SetProp(str(k), str(v))

    return mol
```

[10]:

```
import math

def Rg(filename):
    '''
    Calculates the Radius of Gyration (Rg) of a protein given its .pdb
    structure file. Returns the Rg integer value in Angstrom.
    '''
    coord = list()
    mass = list()
    Structure = open(filename, 'r')
    for line in Structure:
        try:
            line = line.split()
            x = float(line[6])
            y = float(line[7])
            z = float(line[8])
            coord.append([x, y, z])
            if line[-1] == 'C':
                mass.append(12.0107)
            elif line[-1] == 'O':
                mass.append(15.9994)
            elif line[-1] == 'N':
                mass.append(14.0067)
            elif line[-1] == 'S':
                mass.append(32.065)
        except:
            pass
    xm = [(m*i, m*j, m*k) for (i, j, k), m in zip(coord, mass)]
    tmass = sum(mass)
    rr = sum(mi*i + mj*j + mk*k for (i, j, k), (mi, mj, mk) in zip(coord, xm))
    mm = sum((sum(i) / tmass)**2 for i in zip(*xm))
    rg = math.sqrt(rr / tmass-mm)
    return(round(rg, 3))

# Calculate the center based on the reference ligand
Rg = Rg('ligand-{}.pdb'.format(LIGAND_CODE))
print("Radius of Gyration = {}".format(Rg))
```

```
Radius of Gyration = 4.673
```

[11]:

```
# Calculate center of geometry
def COG(pdbfile, include='ATOM,HETATM'):
    """
    Calculates center of geometry of a protein and/or ligand structure.
    Returns:
        center (list): List of float coordinates [x,y,z] that represent the
        center of geometry (precision 3).
    """

    center = [None, None, None]
    include = tuple(include.split(','))

    with open(pdbfile) as pdb:

        # extract coordinates [ [x1,y1,z1], [x2,y2,z2], ... ]
        coordinates = []
        for line in pdb:
            if line.startswith(include):
                coordinates.append([float(line[30:38]),    # x_coord
                                    float(line[38:46]),    # y_coord
                                    float(line[46:54])     # z_coord
                                   ])

        # calculate center of geometry
        center = [sum([coordinates[i][j]/(len(coordinates))
              for i in range(len(coordinates))]) for j in range(3)]
        center = [round(center[i], 3) for i in range(3)]
    return center

CenterOfGeometry = COG('ligand-{}.pdb'.format(LIGAND_CODE))
print("Center Of Geometry = {}".format(CenterOfGeometry))
```

```
Center Of Geometry = [46.929, 12.411, 2.303]
```

[12]:

```
protein = Chem.MolFromPDBFile('{}_prepped.pdb'.format(PDBCODE))
a = MolToPDBQTBlock(protein,flexible=False)
with open('protein.pdbqt','w') as outfile:
    for line in a:
        outfile.write(line)
```

[8]:

```
# You can change the ligand name for each of the generated molecules from your␣
 ↪smiles:
# So, ligand_00 (Should be the reference ligand from the crystal structure)
# Then, ligand_01, ligand_02, ... etc.
# Run one ligand at the time!
LIGAND_TO_DOCK = 'ligand_02' # Write the ligand you want to dock here.
m = Chem.MolFromMolFile('{}.mol'.format(LIGAND_TO_DOCK))
a = MolToPDBQTBlock(m)
with open('{}.pdbqt'.format(LIGAND_TO_DOCK), 'w') as outfile:
    for line in a:
        outfile.write(line)
```

[50]:

```
v = Vina(sf_name='vina')

v.set_receptor('protein.pdbqt')

v.set_ligand_from_file('{}.pdbqt'.format(LIGAND_TO_DOCK))
```

```
center = {
     'center_x' : CenterOfGeometry[0],
     'center_y' : CenterOfGeometry[1],
     'center_z' : CenterOfGeometry[2]

}

size = {
    'size_x' : Rg * 2.9 ,
    'size_y' : Rg * 2.9,
    'size_z' : Rg * 2.9,
           }


v.compute_vina_maps(center=[center['center_x'], center['center_y'],␣
 ↪center['center_z']],
                    box_size=[size['size_x'], size['size_y'], size['size_z']])

'''
# Score the current pose
energy = v.score()
print('Score before minimization: %.3f (kcal/mol)' % energy[0])

# Minimized locally the current pose
energy_minimized = v.optimize()
print('Score after minimization : %.3f (kcal/mol)' % energy_minimized[0])
v.write_pose('1iep_ligand_minimized.pdbqt', overwrite=True)
'''

# Dock the ligand
v.dock(exhaustiveness=10, n_poses=10)
v.write_poses('docking_results.pdbqt', n_poses=1, overwrite=True)
```

```
mode |   affinity | dist from best mode
     | (kcal/mol) | rmsd l.b.| rmsd u.b.
-----+------------+----------+----------
   1       -7.865          0          0
   2       -7.082      3.099      4.413
   3       -6.969      1.577      2.901
   4       -6.925      1.471      2.995
   5       -6.895      3.518      6.909
   6       -6.681      3.269      5.632
   7        -6.47      3.817      8.887
   8       -6.113       2.76      6.023
   9       -6.059      3.401      5.936
```

```
  10         1.611      3.814      5.929
Computing Vina grid … done.
Performing docking (random seed: 2031739639) …
0%   10   20   30   40   50   60   70   80   90   100%
|----|----|----|----|----|----|----|----|----|----|
***************************************************
```

[51]:
```
# Create a mol object from the docking results, write the mol to file and print␣
  ↪the predicted affinity
mol = MolFromPDBQTBlock('docking_results.pdbqt',sanitize=False)
Chem.rdmolfiles.MolToPDBFile(mol, '{}_docked.pdb'.format(LIGAND_TO_DOCK))
Ki = (math.exp(float(mol.GetProp('vina_affinity'))/0.592))
pKi = -math.log10(Ki)
pChEMBL = pKi
print('The affinity predicted by Vina = {} (kcal/mol)'.format(mol.
  ↪GetProp('vina_affinity')))
print('Which (approximately) corresponds to a pChEMBL of {:.2f}'.format(pKi))
```

```
The affinity predicted by Vina = -7.446 (kcal/mol)
Which (approximately) corresponds to a pChEMBL of 5.46
```

[11]:
```
view = py3Dmol.view()
view.removeAllModels()
view.setViewStyle({'style':'outline','color':'black','width':0.1})

view.addModel(open('{}_prepped.pdb'.format(PDBCODE),'r').read(),format='pdb')
Prot=view.getModel()
Prot.setStyle({'cartoon':{'arrows':True, 'tubes':True, 'style':'oval', 'color':
  ↪'white'}})
view.addSurface(py3Dmol.VDW,{'opacity':0.6,'color':'white'})

ligtmp = 'ligand-{}.pdb'.format(LIGAND_CODE)
view.addModels(open(ligtmp,'r').read(),format='pdb')
ref_m = view.getModel()
ref_m.setStyle({},{'stick':{'colorscheme':'magentaCarbon','radius':0.2}})

docktmp = '{}_docked.pdb'.format(LIGAND_TO_DOCK)
view.addModels(open(docktmp,'r').read(),format='pdb')
x = view.getModel()
x.setStyle({},{'stick':{'colorscheme':'cyanCarbon','radius':0.2}})

print('Reference: Magenta | Vina Pose: Cyan')

view.zoomTo()
view.show()
```

```
Reference: Magenta | Vina Pose: Cyan
```

[53]:

```
# combine protein and ligand files
filenames = [
'{}_prepped.pdb'.format(PDBCODE),
'{}_docked.pdb'.format(LIGAND_TO_DOCK)
]
with open('{}-complex.pdb'.format(LIGAND_TO_DOCK), 'w') as outfile:
    for fname in filenames:
        with open(fname) as infile:
            for line in infile:
                if not "END" in line:
                    outfile.write(line)
```

[15]:

```
import nglview
def show_residues_around(view, component_index=0, selection="ligand", radius=5.
 ↪0):
    js = (
        f"""
        // Get first (and only) loaded component: our protein-ligand system
        var system = this.stage.compList[{component_index}];
        // Store current selection, we will need it laer
        var prevSele = system.selection.string;
        // Set selection to our desired ligand
        system.setSelection("{selection}");
        // Select all atoms within 5A from the ligand
        var around = system.structure.getAtomSetWithinSelection(system.
 ↪selection, {radius});
        """
        """
        // Extend selection so it includes full residues
        var around_complete = system.structure.getAtomSetWithinGroup(around);
        // Add representation for those atoms
        system.addRepresentation("licorice", {sele: around_complete.
 ↪toSeleString()});
        // Restore selection to original one; otherwise changes won't be␣
 ↪reflected
        system.setSelection(prevSele)
        """
    )
    view._execute_js_code(js)
with open('{}-complex.pdb'.format(LIGAND_TO_DOCK)) as f:
    view = nglview.show_file(f, ext="pdb")

view.center("UNL")
show_residues_around(view, selection="UNL")
view
```

```
NGLWidget()
```