A Jupyter Notebook to aid Docking to MurD protein

This notebook implements a typical protocol for docking ligands to a target protein. It uses RDKit (http://www.rdkit.org) to generate a number of reasonable conformations for each ligand and then uses SMINA (https://sourceforge.net/projects/smina/) to do the docking. Two methods of docking are implemented, the first docks into a rigid receptor, the second sets the protein side-chains around the active site to be flexible. Bear in mind flexible docking will be much, much slower. In the optional final step the resulting docked poses are rescored using a random forest model described in https://www.nature.com/articles/srep46710.

In [13]:

```
import sys
from collections import defaultdict
import numpy as np
from rdkit import Chem
from rdkit.Chem import AllChem
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem import PandasTools
import pandas as pd
IPythonConsole.ipython_3d=True
%pylab inline

import py3Dmol
```

```
------------------------------------------------------------------------
---
ModuleNotFoundError                       Traceback (most recent call la
st)
<ipython-input-13-2ebce9491ba2> in <module>
      2 from collections import defaultdict
      3 import numpy as np
----> 4 from rdkit import Chem
      5 from rdkit.Chem import AllChem
      6 from rdkit.Chem.Draw import IPythonConsole

ModuleNotFoundError: No module named 'rdkit'
```

File location of structures for docking and file format

First we need get the location of the input file of structures you want to dock, replace "asinexSelection.sdf" with your file. You may want to rename the output file for conformations, and the output file containing the docked structures.

The sdf file needs to have the name included in the first line of each molecule record.

```
AEM 10028511
  MOE2019           2D
```

```
 22 24  0  0  0  0  0  0  0  0999 V2000
    7.2040   -6.7290    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    6.3790   -6.7290    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
```

In [ ]:

```
# File locations
sdfFilePath = 'compoundsfordockingpart2.sdf' # The input file of structur
ConfoutputFilePath = 'compoundsfordockingpart2conformations.sdf' # Output

inputMols = [x for x in Chem.SDMolSupplier(sdfFilePath,removeHs=False)]
# Assign atomic chirality based on the structures:
len(inputMols) # Check how many strucures
```

In [ ]:

```
heck that all molecules have a name
r i, mol in enumerate(inputMols):
  if mol is None:
      print('Warning: Failed to read molecule %s in %s' % (i, sdfFilePath)
  if not mol.GetProp('_Name'):

      print('Warning: No name for molecule %s in %s' % (i, sdfFilePath))
```

Docking to Protein

After generating the conformations we can now do the docking. In this example we use smina which can be downloaded from https://sourceforge.net/projects/smina/ you will need to know where smina has been installed. The protein and ligand examples provided are taken from https://fragalysis.diamond.ac.uk/viewer/react/preview/target/MURD MURD-x0373.

Docking using smina
Need protein minus the ligand in pdb format,
the ligand extracted from binding site in pdb format,
Conformations to be docked as sdf from conformation generation above
DockedFilePath = 'All_Docked.sdf.gz' is the File for the Docked structures

In [ ]:

```


ProteinForDocking = 'nsp13protein_only.pdb'
LigandFromProtein = '5rm3ligand only.pdb'
DockedFilePath = 'All_Docked.sdf.gz'
FlexibleDockedFilePath = 'FlexDocked.sdf.gz'
```

In [14]:

```
!'/usr/local/bin/smina.osx' --cpu 10 --seed 0 --autobox_ligand '{LigandFr
```

```
/bin/bash: /usr/local/bin/smina.osx: No such file or directory
```

Flexible docking method, set all side chains within specified distance to flexdist_ligand to flexible
This will take an order of magnitude longer. Currently disabled, to enable remove the #

In [15]:

```
#!'/usr/local/bin/smina.osx' --cpu 10 --seed 0 --autobox_ligand '{LigandF
```

Rescoring using Random Forest Model

Optional, Rescore using a random forest model described in https://www.nature.com/articles/srep46710

Download from https://github.com/oddt/rfscorevs You will need the path to the binary

Path to protein containing ligand in pdb format
protein_plus_373ligand from Diamond
File to store rescored results

In [16]:

```


TargetProtein = 'protein_and_ligand.pdb'
scoreResults = 'DockedRescored.csv'
```

In [17]:

```
!/usr/local/bin/rf-score-vs --receptor '{TargetProtein}' '{DockedFilePath
```

```
/bin/bash: /usr/local/bin/rf-score-vs: No such file or directory
```

In [ ]:

```

```

In [18]:

```

docked_df = PandasTools.LoadSDF(DockedFilePath,molColName='Molecule', rem
```

```
-------------------------------------------------------------------------
---
NameError                                 Traceback (most recent call la
st)
<ipython-input-18-83f3a24fa746> in <module>
----> 1 docked_df = PandasTools.LoadSDF(DockedFilePath,molColName='Molec
ule', removeHs=False)

NameError: name 'PandasTools' is not defined
```