Load the BINANA module

```
import sys
sys.path.append("../")
import binana
```

Load optional visualization module for this demo

```
import py3Dmol
```

Load in a receptor and ligand

```
ligand, receptor = binana.load_ligand_receptor.from_files("ligand.pdbqt", "receptor.p
dbqt")
```

```
ligand, receptor
```

```
(<binana._structure.mol.Mol at 0x7fd4d81f8470>,
 <binana._structure.mol.Mol at 0x7fd4d81f87f0>)
```

Get information about the hydrogen bonds (example)

```
hbond_inf = binana.interactions.get_hydrogen_bonds(ligand, receptor)
```

```
# Counting/characterizing the acceptors and donors (counts)
hbond_inf["counts"]
```

```
{'HDONOR_LIGAND_SIDECHAIN_OTHER': 1, 'HDONOR_RECEPTOR_SIDECHAIN_OTHER': 1}
```

```
# List the atoms involved in each hydrogen bond, and the angles/distances
for hbond_label in hbond_inf["labels"]:
    print(hbond_label)
```

```
('A:CHT(1):N1(14)', 'A:CHT(1):H1(16)', 'A:ASP(157):OD2(285)', 'LIGAND', {'distance':
2.6500811308335455, 'angle': 16.087842801376098})
('A:CHT(1):O6(22)', 'A:ASN(156):2HD2(276)', 'A:ASN(156):ND2(274)', 'RECEPTOR', {'dist
ance': 2.9006795755477723, 'angle': 35.51562311681741})
```

Get information about the cation-pi interactions (example)

```
cation_pi_inf = binana.interactions.get_cation_pi(ligand, receptor)
```

```
# Counting/characterizing the acceptors and donors (counts)
cation_pi_inf["counts"]
```

```
{'PI-CATION_LIGAND-CHARGED_OTHER': 2,
 'PI-CATION_LIGAND-CHARGED_BETA': 2,
 'PI-CATION_RECEPTOR-CHARGED_OTHER': 1}
```

```
# List the atoms involved in each cation-pi interaction
for cation_pi_label in cation_pi_inf["labels"]:
    print(cation_pi_label)
```

```
('[A:CHT(1):N1(2) / A:CHT(1):C5(1) / A:CHT(1):C6(3) / A:CHT(1):C6(4) / A:CHT(1):C7(9)
]', '[A:TRP(43):CG(28) / A:TRP(43):CD1(29) / A:TRP(43):NE1(31) / A:TRP(43):CE2(32) /
A:TRP(43):CD2(30)]', {'distance': 4.403228947034208})
('[A:CHT(1):N1(2) / A:CHT(1):C5(1) / A:CHT(1):C6(3) / A:CHT(1):C6(4) / A:CHT(1):C7(9)
]', '[A:TRP(43):CE2(32) / A:TRP(43):CD2(30) / A:TRP(43):CE3(33) / A:TRP(43):CZ3(35) /
 A:TRP(43):CH2(36) / A:TRP(43):CZ2(34)]', {'distance': 4.280756595250165})
('[A:CHT(1):N1(2) / A:CHT(1):C5(1) / A:CHT(1):C6(3) / A:CHT(1):C6(4) / A:CHT(1):C7(9)
]', '[A:TRP(205):CG(468) / A:TRP(205):CD1(469) / A:TRP(205):NE1(471) / A:TRP(205):CE2
(472) / A:TRP(205):CD2(470)]', {'distance': 4.1748128341280175})
('[A:CHT(1):N1(2) / A:CHT(1):C5(1) / A:CHT(1):C6(3) / A:CHT(1):C6(4) / A:CHT(1):C7(9)
]', '[A:TRP(205):CE2(472) / A:TRP(205):CD2(470) / A:TRP(205):CE3(473) / A:TRP(205):CZ
3(475) / A:TRP(205):CH2(476) / A:TRP(205):CZ2(474)]', {'distance': 4.45074514048553})
('[A:CHT(1):C2(17) / A:CHT(1):O1(18) / A:CHT(1):C5(19) / A:CHT(1):C4(20) / A:CHT(1):C
3(21)]', '[A:LYS(94):NZ(144) / A:LYS(94):HZ1(146) / A:LYS(94):HZ2(147) / A:LYS(94):HZ
3(148)]', {'distance': 2.57953875721998})
```

Other interactions are also available

```
print("Available functions for detecting interactions:")
for f in dir(binana.interactions):
    if "get_" in f:
        print("    " + f)
```

```
Available functions for detecting interactions:
    get_active_site_flexibility
    get_all_interactions
    get_cation_pi
    get_close
    get_closest
    get_electrostatic_energies
    get_halogen_bonds
    get_hydrogen_bonds
    get_hydrophobics
    get_ligand_atom_types
    get_metal_coordinations
    get_pi_pi
    get_salt_bridges
```

Get and display PDB-formatted text

```
pdb_txt = binana.output.pdb_file.write(
    ligand, receptor,
    hydrogen_bonds=hbond_inf,
    cat_pi=cation_pi_inf,
    as_str=True
)
print(
    "\n".join(
        [
            l
            for l
            in pdb_txt.split("\n")
            if l.startswith("REMARK")
        ]
    )
)
```

```
REMARK
REMARK The residue named "CCN" contains the closest contacts between the
REMARK protein and receptor. "CON" indicates close contacts. "ALP", "BET", and
REMARK "OTH" indicate receptor contacts whose respective protein residues have
REMARK the alpha-helix, beta-sheet, or "other" secondary structure. "BAC" and
REMARK "SID" indicate receptor contacts that are part of the protein backbone
REMARK and sidechain, respectively. "HYD" indicates hydrophobic contacts
REMARK between the protein and ligand. "HBN" indicates hydrogen bonds. "HAL"
REMARK indicates halogen bonds. "SAL" indicates salt bridges. "PIS" indicates
REMARK pi-pi stacking interactions, "PIT" indicates T-stacking interactions,
REMARK and "PIC" indicates cation-pi interactions. "MTL" indicates metal-
REMARK coordination interactions. Protein residue names are unchanged, but the
REMARK ligand residue is now named "LIG".
REMARK
```

```
view = py3Dmol.view(data="",linked=False)

view.addModel(pdb_txt)

view.setStyle({'stick':{'radius':0.1}})
view.setStyle({"resn": "LIG"}, {'stick':{'radius':0.3}})
view.setStyle({"resn": "HBN"}, {'sphere':{'radius':1, "color": "yellow"}})
view.setStyle({"resn": "PIC"}, {'sphere':{'radius':0.5, "color": "pink"}})

view.zoomTo()
```

```
<py3Dmol.view at 0x7fd4e85f8630>
```

Get the interactions as a dictionary for easier big-data analysis

```
data = binana.output.dictionary.collect(
    hydrogen_bonds=hbond_inf,
    cat_pi=cation_pi_inf
)

print("Keys:")
print(list(data.keys()))

print("")
print("Hydrogen-bond data (example):")
print(data["hydrogenBonds"])
```

```
Keys:
['hydrogenBonds', 'cationPiInteractions']

Hydrogen-bond data (example):
[{'ligandAtoms': [{'chain': 'A', 'resID': 1, 'resName': 'CHT', 'atomName': 'N1', 'ato
mIndex': 14}, {'chain': 'A', 'resID': 1, 'resName': 'CHT', 'atomName': 'H1', 'atomInd
ex': 16}], 'receptorAtoms': [{'chain': 'A', 'resID': 157, 'resName': 'ASP', 'atomName
': 'OD2', 'atomIndex': 285}], 'metrics': {'distance': 2.6500811308335455, 'angle': 16
.087842801376098}}, {'ligandAtoms': [{'chain': 'A', 'resID': 1, 'resName': 'CHT', 'at
omName': 'O6', 'atomIndex': 22}], 'receptorAtoms': [{'chain': 'A', 'resID': 156, 'res
Name': 'ASN', 'atomName': 'ND2', 'atomIndex': 274}, {'chain': 'A', 'resID': 156, 'res
Name': 'ASN', 'atomName': '2HD2', 'atomIndex': 276}], 'metrics': {'distance': 2.90067
95755477723, 'angle': 35.51562311681741}}]
```

Some prefer CSV-formatted data

```
print(binana.output.csv.collect(data)[:500] + "\n\n...")
```

```
cationPiInteractions
,cationPiInteractions.1
,,ligandAtoms
,,,ligandAtoms.1
,,,,atomIndex,2
,,,,atomName,N1
,,,,chain,A
,,,,resID,1
,,,,resName,CHT
,,,ligandAtoms.2
,,,,atomIndex,1
,,,,atomName,C5
,,,,chain,A
,,,,resID,1
,,,,resName,CHT
,,,ligandAtoms.3
,,,,atomIndex,3
,,,,atomName,C6
,,,,chain,A
,,,,resID,1
,,,,resName,CHT
,,,ligandAtoms.4
,,,,atomIndex,4
,,,,atomName,C6
,,,,chain,A
,,,,resID,1
,,,,resName,CHT
,,,ligandAtoms.5
,,,,atomIndex,9
,,,,atomName,C7
,,,,chain,A
,,,,resID,1
,,,,resName,

...
```

Get all the interactions at once

```
all_inf = binana.interactions.get_all_interactions(ligand, receptor)
```

```
all_inf.keys()
```

```
dict_keys(['closest', 'close', 'electrostatic_energies', 'active_site_flexibility', '
hydrophobics', 'hydrogen_bonds', 'halogen_bonds', 'ligand_atom_types', 'pi_pi', 'cat_
pi', 'salt_bridges', 'metal_coordinations', 'ligand_rotatable_bonds'])
```

Get and display PDB-formatted text containing all interactions

```
pdb_txt = binana.output.pdb_file.write_all(
    ligand, receptor,
    all_inf,
    None,
    as_str=True
)
```

```
view = py3Dmol.view(data="",linked=False)

view.addModel(pdb_txt)

view.setStyle({'stick':{'radius':0.1}})
view.setStyle({"resn": "LIG"}, {'stick':{'radius':0.3}})
view.setStyle({"resn": "CCN"}, {'sphere':{'radius':0.5, "color": "yellow"}})
view.setStyle({"resn": "SAL"}, {'sphere':{'radius':1, "color": "pink"}})

view.zoomTo()
```

```
<py3Dmol.view at 0x7fd4e86ab7b8>
```

Get all interactions as a single dictionary

```
all_data = binana.output.dictionary.collect_all(all_inf)

print(all_data.keys())
print()
print("Hydrogen bonds (example):")
print()
print(all_data["hydrogenBonds"])
```

```
dict_keys(['closestContacts', 'closeContacts', 'hydrophobicContacts', 'hydrogenBonds'
, 'piPiStackingInteractions', 'tStackingInteractions', 'cationPiInteractions', 'saltB
ridges', 'activeSiteFlexibility', 'electrostaticEnergies', 'ligandAtomTypes', 'ligand
RotatableBonds'])

Hydrogen bonds (example):

[{'ligandAtoms': [{'chain': 'A', 'resID': 1, 'resName': 'LIG', 'atomName': 'N1', 'ato
mIndex': 14}, {'chain': 'A', 'resID': 1, 'resName': 'LIG', 'atomName': 'H1', 'atomInd
ex': 16}], 'receptorAtoms': [{'chain': 'A', 'resID': 157, 'resName': 'ASP', 'atomName
': 'OD2', 'atomIndex': 285}], 'metrics': {'distance': 2.6500811308335455, 'angle': 16
.087842801376098}}, {'ligandAtoms': [{'chain': 'A', 'resID': 1, 'resName': 'LIG', 'at
omName': 'O6', 'atomIndex': 22}], 'receptorAtoms': [{'chain': 'A', 'resID': 156, 'res
Name': 'ASN', 'atomName': 'ND2', 'atomIndex': 274}, {'chain': 'A', 'resID': 156, 'res
Name': 'ASN', 'atomName': '2HD2', 'atomIndex': 276}], 'metrics': {'distance': 2.90067
95755477723, 'angle': 35.51562311681741}}]
```