```
#install the ChEMBL database
#ChEMBL is a manually curated database of bioactive molecules with drug-like properties.
#It brings together chemical, bioactivity, and genomic data to aid the translation of genomic information into effecti
! pip install chembl_webresource_client



#accessing the database
import pandas as pd
from chembl_webresource_client.new_client import new_client


#searching the database - here we have chosen coronavirus
target = new_client.target
target_query = target.search('coronavirus')
targets = pd.DataFrame.from_dict(target_query)
targets
```

| | cross_references | organism | pref_name | score | species_group_flag | target_chembl_id | target_components | ta |
|---|---|---|---|---|---|---|---|---|
| **0** | [] | Coronavirus | Coronavirus | 17.0 | False | CHEMBL613732 | [] | |
| **1** | [] | SARS coronavirus | SARS coronavirus | 14.0 | False | CHEMBL612575 | [] | |
| **2** | [] | Feline coronavirus | Feline coronavirus | 14.0 | False | CHEMBL612744 | [] | |
| **3** | [] | Murine coronavirus | Murine coronavirus | 14.0 | False | CHEMBL5209664 | [] | |
| **4** | [] | Human coronavirus 229E | Human coronavirus 229E | 12.0 | False | CHEMBL613837 | [] | |
| **5** | [] | Human coronavirus OC43 | Human coronavirus OC43 | 12.0 | False | CHEMBL5209665 | [] | |
| **6** | [{'xref_id': 'P0C6U8', 'xref_name': None, 'xre... | SARS coronavirus | SARS coronavirus 3C-like proteinase | 10.0 | False | CHEMBL3927 | [{'accession': 'P0C6U8', 'component_descriptio... | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **7** | | [] | Middle East respiratory syndrome-related coron... | Middle East respiratory syndrome-related coron... | 9.0 | | False | CHEMBL4296578 | [] |
| **8** | [{'xref_id': 'P0C6X7', 'xref_name': None, 'xre... | | SARS coronavirus | Replicase polyprotein 1ab | 4.0 | | False | CHEMBL5118 | [{'accession': 'P0C6X7', 'component_descriptio... |
| **9** | | [] | Severe acute respiratory syndrome coronavirus 2 | Replicase polyprotein 1ab | 4.0 | | False | CHEMBL4523582 | [{'accession': 'P0DTD1', 'component_descriptio... |

Next steps: Generate code with `targets` View recommended plots

```
#this project we are interested in target types particulary single protein
#we are going to investigate further SARS coronavirus 3C-like proteinase
selected_target = targets.target_chembl_id[6]
selected_target
```

```
'CHEMBL3927'
```

```
#here we filter off the standard type
#the standard type is the measured property or activity of a chemical compound
#here we use half maximal inhibitory concentration (IC50)
activity = new_client.activity
res = activity.filter(target_chembl_id=selected_target).filter(standard_type='IC50')
```

```
#this dataframe shows us key values
#for example standard value which tells us the needed concentration of the compound in order to reach half maximal in
#the lower the standard value the less of the compound that is needed
df = pd.DataFrame.from_dict(res)
df
```

| | action_type | activity_comment | activity_id | activity_properties | assay_ |
|---|---|---|---|---|---|
| **0** | None | None | 1480935 | [] | CHE |
| **1** | None | None | 1480936 | [] | CHE |
| **2** | None | None | 1481061 | [] | CHE |
| **3** | None | None | 1481065 | [] | CHE |
| **4** | None | None | 1481066 | [] | CHE |
| **...** | ... | ... | ... | ... | |
| **128** | None | None | 12041507 | [] | CHEM |
| **129** | None | None | 12041508 | [] | CHEM |
| **130** | None | None | 12041509 | [] | CHEM |
| **131** | None | None | 12041510 | [] | CHEM |
| **132** | None | None | 12041511 | [] | CHEM |

133 rows × 46 columns

```
#save our pulled data into a csv file
df.to_csv('Coronavirus_bioactivity_data.csv', index=False)


Bio_act_data = pd.read_csv('Coronavirus_bioactivity_data.csv')


! head Coronavirus_bioactivity_data.csv
```

```
activity_comment,activity_id,activity_properties,assay_chembl_id,assay_description,assay_type,assay_variant_acces
,CHEMBL829584,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
,CHEMBL829584,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
,CHEMBL830868,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
,CHEMBL829584,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
,CHEMBL829584,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
,CHEMBL828143,In vitro inhibitory concentration SARS coronavirus main protease (SARS CoV 3C-like protease) ,B,,,BA
,CHEMBL829584,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
,CHEMBL829584,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
,CHEMBL829584,In vitro inhibitory concentration against SARS coronavirus main protease (SARS CoV 3C-like protease
```

```
Bio_act_data.standard_value.count()
```

```
133
```

```
#evaluate for missing data
#none in this sample
print(Bio_act_data.standard_value.notna().sum())
```

```
133
```

```
#we are now going to create classes to define the compounds bioactivity off of the standard_value (nM)
bioactivity_class = []
for i in Bio_act_data.standard_value:
  if float(i) >= 10000:
    bioactivity_class.append('inactive')
  elif float(i) <= 1000:
    bioactivity_class.append('active')
```

```
  else:
    bioactivity_class.append('indeterminate')


#iterate the molecule ChEMBL id to a list
#here we are labeling the molecules that have a modulatory effect on the target protein
#we are doing this to remove redundacy
mol_cid = []
for i in Bio_act_data.molecule_chembl_id:
  mol_cid.append(i)


#simplified molecular input line entry system
#unique textual representation of a molecular structure
canonical_smiles = []
for i in Bio_act_data.canonical_smiles:
  canonical_smiles.append(i)


standard_value = []
for i in Bio_act_data.standard_value:
  standard_value.append(i)


data_tuples = list(zip(mol_cid, canonical_smiles, bioactivity_class, standard_value))
df2 = pd.DataFrame(data_tuples, columns=['molecule_chembl_id', 'canonical_smiles', 'bioactivity_class', 'standard_val


df2.head(5)
```

| | molecule_chembl_id | canonical_smiles | bioactivity_cla |
|---|---|---|---|
| **0** | CHEMBL187579 | Cc1noc(C)c1CN1C(=O)C(=O)c2cc(C#N)ccc21 | indetermin |
| **1** | CHEMBL188487 | O=C1C(=O)N(Cc2ccc(F)cc2Cl)c2ccc(I)cc21 | indetermin |
| **2** | CHEMBL185698 | O=C1C(=O)N(CC2COc3ccccc3O2)c2ccc(I)cc21 | inac |
| **3** | CHEMBL426082 | O=C1C(=O)N(Cc2cc3ccccc3s2)c2ccccc21 | inac |
| **4** | CHEMBL187717 | O=C1C(=O)N(Cc2cc3ccccc3s2)c2c1cccc2[N+] | indetermin |

Next steps: Generate code with df2 View recommended plots

```
df2.to_csv('Coronavirus_bioactivity_data_preprocessed.csv', index=False)
```

```
! ls -1
```

```
Coronavirus_bioactivity_data.csv
Coronavirus_bioactivity_data_preprocessed.csv
sample_data
```