Get Pubchem drug feautures

1. finding corresponding Pubchem ids for the drugs
2. call Pubchem to get chemical properties of the drugs
3. Preprocess text Drug description from the original datasets
4. Preprocess some text characteristics from PubChem properties

In [1]:

```
import pandas as pd
import numpy as np
import os
# pip install PubChemPy
import pubchempy as pcp
import re
from pubchempy import Compound
import warnings
warnings.filterwarnings("ignore")
import time
from tqdm import tqdm
import pickle
from functions.drug_features import *

_FOLDER = "database/"
_FOLDER_2 = "results/"
```

In [2]:

```
drug_features = pd.read_csv(_FOLDER + "Drug_Features.csv").rename(columns={"Drug ID": "DRUG_ID",
                                                                           "Drug Name": "Drug_Nam
                                                                          "Target Pathway": "Targ
#drug_features.set_index("DRUG_ID", inplace= True)
df = pd.read_csv(_FOLDER+'Drug_Features.csv')
ind_wrong_name = df[df["Drug Name"]=="Lestauritinib"].index
df.loc[ind_wrong_name, "Drug Name"] = "Lestaurtinib"
print(drug_features.shape)
drug_features.head()
```

(265, 5)

Out[2]:

	DRUG_ID	Drug_Name	Synonyms	Target	Target_Pathway
0	1	Erlotinib	Tarceva, RG-1415, CP-358774, OSI-774, Ro-50823...	EGFR	EGFR signaling
1	3	Rapamycin	AY-22989, Sirolimus, WY-090217, Torisel, Rapamune	MTORC1	PI3K/MTOR signaling
2	5	Sunitinib	Sutent, Sunitinib Malate, SU-11248	PDGFR, KIT, VEGFR, FLT3, RET, CSF1R	RTK signaling
3	6	PHA-665752	PHA665752, PHA 665752	MET	RTK signaling
4	9	MG-132	LLL cpd, MG 132, MG132	Proteasome, CAPN1	Protein stability and degradation

Part 1: Get PubChem ids

In [14]:

```
#the most recent data on drugs properties
#https://www.cancerrxgene.org/downloads/anova?screening_set=GDSC1
drugs_1 = pd.read_csv(_FOLDER+'drugs_gdsc1.csv')
drug_features = pd.read_csv(_FOLDER + "Drug_Features.csv").rename(columns={"Drug ID": "DRUG_ID",
                                                                           "Drug Name": "Drug_Nam
                                                                          "Target Pathway": "Targ

drug_names_dict = dict(zip(drug_features["DRUG_ID"], drug_features["Drug_Name"]))
```

In [15]:

```
drugs_1.head()
```

Out[15]:

	drug_id	drug_name	synonyms	pathway_name	targets	pubchem
0	1170	CCT-018159	CCT018159, CCT 018159	Protein stability and degradation	HSP90	5327091
1	1198	FY012	-	Unclassified	-	-
2	1251	N22899-6-C1	-	Unclassified	Peptidyl arginine deaminase (PAD)	-
3	1259	Talazoparib	BMN-673, BMN 973	Genome integrity	PARP1, PARP2	44819241
4	1359	965-D2	-	Other, kinases	MKNK1, MKNK2, PIM1, PIM3	-

In [16]:

```
drug_features.head()
```

Out[16]:

	DRUG_ID	Drug_Name	Synonyms	Target	Target_Pathway
0	1	Erlotinib	Tarceva, RG-1415, CP-358774, OSI-774, Ro-50823...	EGFR	EGFR signaling
1	3	Rapamycin	AY-22989, Sirolimus, WY-090217, Torisel, Rapamune	MTORC1	PI3K/MTOR signaling
2	5	Sunitinib	Sutent, Sunitinib Malate, SU-11248	PDGFR, KIT, VEGFR, FLT3, RET, CSF1R	RTK signaling
3	6	PHA-665752	PHA665752, PHA 665752	MET	RTK signaling
4	9	MG-132	LLL cpd, MG 132, MG132	Proteasome, CAPN1	Protein stability and degradation

In [17]:

```
# drug_ids are not the same in both datasets,
# but we can take mapping of drugname and pubchemid

drugs_1["pubchem"].value_counts().head()
```

Out[17]:

```
-           137
none         25
several       3
9956119       2
46224516      2
Name: pubchem, dtype: int64
```

In [18]:

```
# several drug_id can refer to the same drug with the same pubchemid
drugs_1[drugs_1["pubchem"]=="46907787"]
```

Out[18]:

	drug_id	drug_name	synonyms	pathway_name	targets	pubchem
301	1218	JQ1	JQ-1, (+)-JQ-1	Chromatin other	BRD2, BRD3, BRD4, BRDT	46907787
380	163	JQ1	JQ-1, (+)-JQ-1	Chromatin other	BRD2, BRD3, BRD4, BRDT	46907787

In [19]:

```
df = drugs_1[(drugs_1["pubchem"]!= "-") & (drugs_1["pubchem"] != "none") & (drugs_1["pubchem"] !
print("All drugs in gdsc1: %d, With pubchem_ids: %d" % (drugs_1.shape[0], df.shape[0]))

print("Number of drugs in drug_features:", drug_features.shape[0])

# match drug name and pubchem_id based on availble data
drug_pubchem_dict = dict(zip(df["drug_name"], df["pubchem"]))
drug_features["pubchem_id"] = drug_features["Drug_Name"].map(drug_pubchem_dict)

print("Number of missing data:", drug_features["pubchem_id"].isnull().sum())

# run function calling pubchem to get pubchem ids for drug names
missing_drug_ids = drug_features[drug_features["pubchem_id"].isnull()]["DRUG_ID"].values
pubchem_ids_good, pubchem_ids_bad = get_pubchem_id(missing_drug_ids, drug_names_dict)
```

```
All drugs in gdsc1: 403, With pubchem_ids: 238
Number of drugs in drug_features: 265
Number of missing data: 33

100%|██████████████████████████████████████████████████████████████████████████
██████| 33/33 [00:14<00:00,  2.26it/s]
```

In [20]:

```
# run corrections
print("Number of found ids %d, Number of not found: %d" % (len(pubchem_ids_good), len(pubchem_ids

if "DRUG_ID" in drug_features.columns:
    drug_features.set_index("DRUG_ID", inplace = True)

for drug_id in pubchem_ids_good:
    drug_features.loc[drug_id, "pubchem_id"] = pubchem_ids_good[drug_id]
```

```
Number of found ids 8, Number of not found: 25
```

In [21]:

```
pubchem_ids_good, not_identified_drugs = run_manual_corrections(pubchem_ids_bad, drop_not_found_c

for drug_id in pubchem_ids_good:
    drug_features.loc[drug_id, "pubchem_id"] = pubchem_ids_good[drug_id]

print("WnNumber of found ids found after manual corrections %d, Number of not found: %d" % (len(p
```

```
Total number of drugs: 25
Total number of drugs for correction: 25
Number of corrected drugs: 10
Number of not found drugs: 15
Final number of drugs with PubChem id: 25

Number of found ids found after manual corrections 10, Number of not found: 15
```

In [22]:

```
drug_features[drug_features["pubchem_id"].isnull()]
```

Out[22]:

DRUG_ID	Drug_Name	Synonyms	Target	Target_Pathway	pubchem_id
164	JQ12	-	HDAC1, HDAC2	Chromatin histone acetylation	NaN
211	TL-2-105	-	not defined	ERK MAPK signaling	NaN
225	Genentech Cpd 10	-	AURKA, AURKB	Mitosis	NaN
253	XMD14-99	-	ALK, CDK7, LTK, others	Other	NaN
261	TL-1-85	-	TAK	Other, kinases	NaN
286	KIN001-236	-	Angiopoietin-1 receptor	Other	NaN
330	XMD13-2	-	RIPK1	Apoptosis regulation	NaN
332	XMD15-27	-	CAMK2	Other, kinases	NaN
1037	BX796	BX-796	TBK1, PDK1 (PDPK1), IKK, AURKB, AURKC	Other	NaN
1142	HG-5-113-01	-	LOK, LTK, TRCB, ABL(T315I)	Other	NaN
1143	HG-5-88-01	-	EGFR, ADCK4	Other, kinases	NaN
1158	XMD11-85h	-	BRSK2, FLT4, MARK4, PRKCD, RET, SPRK1	Other	NaN
1166	QL-VIII-58	-	MTOR, ATR	Other	NaN
1203	QL-XII-61	-	BMX, BTK	Other, kinases	NaN
1261	rTRAIL	-	TRAIL receptor agonist	Apoptosis regulation	NaN

In [23]:

```
drug_features = drug_features[~drug_features["pubchem_id"].isnull()]
drug_features.shape
```

Out[23]:

```
(250, 5)
```

15 drugs from 265 can't be assesed and will not be used in further modelling

Part 2 :Getting properties by PubChem API

In [24]:

```
%%time
drug_features, drugs_to_drop = get_pubchem_properties(drug_features)
```

```
100%|████████████████████████████████████████████████████████████
████| 250/250 [02:03<00:00,  2.02it/s]

CPU times: total: 6.52 s
Wall time: 2min 3s
```

In [25]:

```
len(drugs_to_drop)
```

Out[25]:

```
0
```

In [26]:

```
drug_features.drop(drugs_to_drop, axis=0).to_csv(_FOLDER_2 + "drug_features_raw.csv")
```

End of Part 2: Read prepared data

In [27]:

```
drug_features = pd.read_csv(_FOLDER_2+ "drug_features_raw.csv").set_index("DRUG_ID")
print(drug_features.shape)
```

```
(250, 26)
```

In [28]:

```
# pubchem_id is none

drug_features[drug_features["molecular_weight"].isnull()].shape[0]
```

Out[28]:

```
0
```

In [29]:

```
drug_features.columns
```

Out[29]:

```
Index(['Drug_Name', 'Synonyms', 'Target', 'Target_Pathway', 'pubchem_id',
       'molecular_weight', 'elements', '2bonds', '3bonds', 'xlogp',
       'formal_charge', 'surface_area', 'complexity', 'h_bond_donor_count',
       'h_bond_acceptor_count', 'rotatable_bond_count', 'heavy_atom_count',
       'atom_stereo_count', 'defined_atom_stereo_count',
       'undefined_atom_stereo_count', 'bond_stereo_count',
       'covalent_unit_count', 'molecular_formula', 'canonical_smiles',
       'inchi_string', 'inchi_key'],
      dtype='object')
```

In [30]:

```
int_columns = ['2bonds', '3bonds', 'h_bond_donor_count',
       'h_bond_acceptor_count', 'rotatable_bond_count', 'heavy_atom_count',
       'atom_stereo_count', 'defined_atom_stereo_count',
       'undefined_atom_stereo_count', 'bond_stereo_count',
       'covalent_unit_count']
for col in int_columns:
    drug_features[col] = np.int16(drug_features[col])
```

Part 3: Preprocessing Text PubChem characteristics

Presence of some elements (11 elements)

In [31]:

```
%%time

all_elements = list(set(drug_features["elements"].str.split(",", expand=True).fillna(0).values.f
all_elements

elements_in_drugs= list(set([atom.strip(" ").strip("'") for atom in all_elements]))
exceptions =[]
for drug_index in drug_features.index:
    compound_elements = drug_features.loc[drug_index, "elements"]
#     print(compound_elements)
    try:
        for i, atom in list(enumerate(elements_in_drugs)):
            if atom in compound_elements:
                drug_features.loc[drug_index, atom] = 1
#                 print(atom, "Yes")
            else:
                drug_features.loc[drug_index, atom] = 0
#                 print(atom, "No")
    except:
        exceptions.append(drug_index)
        drug_features.loc[drug_index, atom] = 0

for col in ['B', 'I', 'Br', 'Cl', 'O', 'N', 'F', 'P', 'S', 'Pt']:
    drug_features[col]= np.int16(drug_features[col])

print("Exceptions:", drug_features.loc[exceptions, :].shape[0])
print("Elements in drugs:", len(elements_in_drugs), elements_in_drugs)
```

```
Exceptions: 0
Elements in drugs: 11 ['Pt', 'N', 'Cl', 'I', 'H', 'P', 'O', 'Br', 'F', 'S', 'B']
CPU times: total: 219 ms
Wall time: 212 ms
```

In [32]:

```
drug_features["Br"].value_counts()
```

Out[32]:

```
0    243
1      7
Name: Br, dtype: int64
```

In [33]:

```
drug_features.to_csv(_FOLDER_2 + "drug_features_with_pubchem_properties.csv")

PubChem_features = ["molecular_weight","2bonds", "3bonds", "xlogp", "formal_charge",
    "surface_area", "complexity", "h_bond_donor_count",
    "h_bond_acceptor_count", "rotatable_bond_count",
    "heavy_atom_count", "atom_stereo_count", "defined_atom_stereo_count",
    "undefined_atom_stereo_count", "bond_stereo_count", "covalent_unit_count",
    'B', 'I', 'Br', 'Cl', 'O', 'N', 'F', 'P', 'S', 'Pt']

with open(_FOLDER_2 + "X_PubChem_properties.txt", 'w') as f:
    for s in PubChem_features:
        f.write(str(s) + '₩n')

print("Number of PubChem features:", len(PubChem_features))
```

```
Number of PubChem features: 26
```

End of Part 3: Read Prepared Data

In [34]:

```
drug_features = pd.read_csv(_FOLDER_2+ "drug_features_with_pubchem_properties.csv").set_index("DF

with open(_FOLDER_2 + "X_PubChem_properties.txt", 'r') as f:
    X_PubChem_properties = [line.rstrip('₩n') for line in f]
```

Part 4: Preprocessing Drugs description from original data

In this section, we are going to have some dumnies columns for Target and Target_Pathway

Converting of Target Pathway resulted in 26 new columns

It is also worth considering elements columns and that deleting columns with C and H which are present in all the compounds

Dumnies for Target (229) and Target_Pathway (23)

In [35]:

```
drug_features.head(3)
```

Out[35]:

DRUG_ID	Drug_Name	Synonyms	Target	Target_Pathway	pubchem_id	molecular_weight
1	Erlotinib	Tarceva, RG-1415, CP-358774, OSI-774, Ro-50823...	EGFR	EGFR signaling	176870	393.40
3	Rapamycin	AY-22989, Sirolimus, WY-090217, Torisel, Rapamune	MTORC1	PI3K/MTOR signaling	5384616	184.18
5	Sunitinib	Sutent, Sunitinib Malate, SU-11248	PDGFR, KIT, VEGFR, FLT3, RET, CSF1R	RTK signaling	5329102	398.50

3 rows × 37 columns

In [36]:

```
targets = ""
for x in drug_features["Target"].values:
    targets = targets + ", " + x
targets = list(set(targets.split(", ")[1:]))
print("Number of targets:", len(targets))
```

```
Number of targets: 212
```

In [37]:

```
df_target = pd.DataFrame(data = np.int32(np.zeros([drug_features.shape[0], len(targets)])),
                         index = drug_features.index,
                         columns = targets)
len(targets)
```

Out[37]:

```
212
```

In [38]:

```
for index in drug_features.index:
    targets_i = drug_features.loc[index, "Target"].split(", ")
    df_target.loc[index, targets_i]=1
df_target.shape
```

Out[38]:

```
(250, 213)
```

In [39]:

```
print("Number of unique pathways:", drug_features["Target_Pathway"].nunique())

df_target_target_pathway = pd.concat([df_target, pd.get_dummies(drug_features["Target_Pathway"])]
df_target_target_pathway.shape
```

```
Number of unique pathways: 23
```

Out[39]:

```
(250, 236)
```

In [40]:

```
drug_features["Target_Pathway"]
```

Out[40]:

```
DRUG_ID
1                          EGFR signaling
3                     PI3K/MTOR signaling
5                           RTK signaling
6                           RTK signaling
9       Protein stability and degradation
                      ...
1498                   ERK MAPK signaling
1502                      Hormone-related
1526                   ERK MAPK signaling
1527                  PI3K/MTOR signaling
1529                                Other
Name: Target_Pathway, Length: 250, dtype: object
```

In [41]:

```
df_final = pd.concat([drug_features.drop(["Target_Pathway"], axis=1), df_target_target_pathway],
df_final.shape
```

Out[41]:

```
(250, 272)
```

In [42]:

```
df_final
```

Out[42]:

DRUG_ID	Drug_Name	Synonyms	Target	pubchem_id	molecular_weight	elements	2
1	Erlotinib	Tarceva, RG-1415, CP-358774, OSI-774, Ro-50823...	EGFR	176870	393.40	'C', 'H', 'O', 'N'	
3	Rapamycin	AY-22989, Sirolimus, WY-090217, Torisel, Rapamune	MTORC1	5384616	184.18	'C', 'H', 'N'	
5	Sunitinib	Sutent, Sunitinib Malate, SU-11248	PDGFR, KIT, VEGFR, FLT3, RET, CSF1R	5329102	398.50	'C', 'N', 'H', 'O', 'F'	
6	PHA-665752	PHA665752, PHA 665752	MET	10461815	641.60	'C', 'N', 'Cl', 'H', 'O', 'S'	
9	MG-132	LLL cpd, MG 132, MG132	Proteasome, CAPN1	462382	475.60	'C', 'H', 'O', 'N'	
...	...	...	...	...	...	...	
1498	Selumetinib	AZD6244, AZD-6244, ARRY-886	MEK1, MEK2	10127622	457.70	'C', 'N', 'Cl', 'H', 'O', 'Br', 'F'	
1502	Bicalutamide	ICI-176334, Casodex, Cosudex, ICI 176334	AR	2375	430.40	'C', 'N', 'H', 'O', 'F', 'S'	
1526	Refametinib	RDEA119, BAY-86-9766, BAY 869766	MEK1, MEK2	44182295	572.30	'C', 'I', 'N', 'H', 'O', 'F', 'S'	
1527	Pictilisib	GDC-0941, GDC0941, RG-7621	PI3K (class 1)	17755052	513.60	'C', 'N', 'H', 'O', 'S'	
1529	Pevonedistat	MLN4924, MLN 4924, MLN-4924	NAE	16720766	443.50	'C', 'N', 'H', 'O', 'S'	

250 rows × 272 columns

In [43]:

```
df_final.to_csv(_FOLDER_2 + "drug_features_with_pubchem_properties_final.csv")

with open(_FOLDER_2 + "X_features_Targets.txt", 'w') as f:
    for s in targets:
        f.write(str(s) + '₩n')

with open(_FOLDER_2 + "X_features_Target_Pathway.txt", 'w') as f:
    for s in drug_features["Target_Pathway"].unique():
        f.write(str(s) + '₩n')
```

End of Part 4: Read Prepared Data

In [44]:

```
drug_features = pd.read_csv(_FOLDER_2 + "drug_features_with_pubchem_properties_final.csv")

with open(_FOLDER_2 + "X_features_Targets.txt", 'r') as f:
    X_features_Targets = [line.rstrip('₩n') for line in f]

with open(_FOLDER_2 + "X_features_Target_Pathway.txt", 'r') as f:
    X_features_Target_Pathway = [line.rstrip('₩n') for line in f]
```

In [45]:

```
drug_features.head()
```

Out[45]:

	DRUG_ID	Drug_Name	Synonyms	Target	pubchem_id	molecular_weight	elements
0	1	Erlotinib	Tarceva, RG-1415, CP-358774, OSI-774, Ro-50823...	EGFR	176870	393.40	'C', 'H', 'O', 'N
1	3	Rapamycin	AY-22989, Sirolimus, WY-090217, Torisel, Rapamune	MTORC1	5384616	184.18	'C', 'H', 'N
2	5	Sunitinib	Sutent, Sunitinib Malate, SU-11248	PDGFR, KIT, VEGFR, FLT3, RET, CSF1R	5329102	398.50	'C', 'N', 'H', 'O', 'F
3	6	PHA-665752	PHA665752, PHA 665752	MET	10461815	641.60	'C', 'N', 'Cl', 'H', 'O', 'S
4	9	MG-132	LLL cpd, MG 132, MG132	Proteasome, CAPN1	462382	475.60	'C', 'H', 'O', 'N

5 rows × 273 columns

In [49]:

```
with open(_FOLDER_2+"X_features_cancer_cell_lines.txt", 'r') as f:
    X_cancer_cell_lines = [line.rstrip('₩n') for line in f]
```

In [50]:

```
print("Final Features: ₩n")
print("Cell lines (CCL) features:", len(X_cancer_cell_lines))
print("PubChem drug features:", len(PubChem_features))
print("Drug description features - Targets: %d, Target_Pathway: %d" % (len(X_features_Targets), l
```

```
Final Features:

Cell lines (CCL) features: 1073
PubChem drug features: 26
Drug description features - Targets: 212, Target_Pathway: 23
```

In [51]:

```
all_elements
```

Out[51]:

```
["'F'",
 " 'S'",
 " 'N'",
 " 'Cl'",
 " 'B'",
 "'H'",
 " 'Br'",
 " 'P'",
 " 'I'",
 " 'O'",
 " 'F'",
 " 'Pt'"]
```

In [ ]:

```

```