HEADER

the python package and function in my lib to upload and constants

In [ ]:
```
import pandas as pd
import numpy as np
import networkx as nx
from lib import ReactionEnthalpyCalculation, FromNetworkDataFrameToNetworkCla

################################################################ CONSTANT ###
ph_list = ['e-','CRP','CR','Photon']
nan = ['NaN','nan',np.nan,' ','']
FROMHETOKJMOL = 2625.5002
FROMKTOKJMOL = 0.00831441
############################################################################
```

PARAMETERS THAT MUST BE DEFINED FOR THE CLEANING PROCEDURE

The path_to_network must be in a different folder then the original one, in order to save the information in a work directory and to not override the info.

- path_to_molecules_database --> path to the molecules database
- path_to_network --> path to the network to be clean
- threshold_endothermicity_kJmol --> threshold to determined the reactions to be deleted for endothermicity (in kJ/mol)

In [ ]:
```
path_to_molecules_database = '/Users/tinaccil/Documents/GitHub/GreToBaPe_Clea
path_to_network = '/Users/tinaccil/Documents/GitHub/GreToBaPe_Cleaning/tmp_ne
threshold_endothermicity_kJmol = 10
```

Reading the network and print the species that are not present in the Database Be aware that in many network, the same species is reported with different name. Please control, in the database is reported the number of atoms per species.

In [ ]:
```
with open(path_to_molecules_database,"r") as data:
    df_info_mol = pd.read_csv(data, delimiter = '\t')
#with open(path_to_network,"r") as data:
#    df_net = pd.read_csv(data, delimiter = '\t')
df_net = FromNetworkKidaDATtoCSV(path_to_network,save=False)
df_net_tmp = df_net.copy()
df_mol_new = ExtractMolFromNetwork(df_net)
df_info_mol['In_Network'] = df_info_mol['species'].isin(df_mol_new['species']
df_mol_new['In_database']  = df_mol_new['species'].isin(df_info_mol['species'
df_mol_new = df_mol_new[(~df_mol_new['species'].isin(nan)) | (~df_mol_new['sp
df_info_mol = df_info_mol[df_info_mol['In_Network'] == True]
print("Species not prensent in Database:")
with pd.option_context('display.max_rows', None, 'display.max_columns', None)
    display(df_mol_new[(df_mol_new['In_database']==False) & (~ df_mol_new['sp
del df_mol_new
```

ENDOTHERMICITY CALCULATION

it will generete a `_endo.dat` file with all the reactions adresses as endothermic (i.e. above the threshold_endothermicity_kJmol)

In [ ]:
```
net  = FromNetworkDataFrameToNetworkClassList(df_net_tmp)
df_info_mol['Energy'].astype(float)
df_info_mol['ZPE'].astype(float)
df_info_mol['enthalpy'] = df_info_mol['Energy'] + df_info_mol['ZPE']
df_net_tmp['enthalpy'] = np.array([ReactionEnthalpyCalculation(net,df_info_mo
df_net_tmp['enthalpy'] = df_net_tmp['enthalpy'] * FROMHETOKJMOL
print('tot reaction with enthalpy data: ' + str(df_net_tmp[df_net_tmp['enthal
#with open("/Users/tinaccil/Documents/GitHub/GreToBaPe_Cleaning/tmp_network_t
#    output.write(df_net_tmp.to_csv(sep="\t", index=False))
tmp_endo = df_net_tmp[df_net_tmp['enthalpy'].notnull()].copy()
print('tot reaction endothermic: ' + str(tmp_endo[tmp_endo['enthalpy'] > thre
print('tot reaction endothermic not formula 3: ' + str(tmp_endo[(tmp_endo['en
#print('tot reaction endothermic formula 3 with dH/C > ' + str(epsilon) + ':
print('tot reaction endothermic formula 3 with dH-C > ' + str(threshold_endot
print('tot reaction endothermic formula 3 barrierless: ' + str(tmp_endo[(tmp_
tmp_endo = tmp_endo[((tmp_endo['enthalpy'] > 0) & (tmp_endo['enthalpy'] < thr
#tmp_endo = tmp_endo[((tmp_endo['enthalpy'] > threshold_endothermicity_kJmol)
tmp_endo['C_kjmol'] = tmp_endo['C']*FROMKTOKJMOL
tmp_endo['enthalpy-C_kjmol'] = tmp_endo['enthalpy'] - (tmp_endo['C']*FROMKTOK
print('tot reaction to be deleted: ' + str(tmp_endo.shape[0]))
with pd.option_context('display.max_rows', None, 'display.max_columns', None)
    display(tmp_endo)
FromNetworkCSVtoKidaDAT(tmp_endo,path_to_network[:-4] + '_endo.dat')
with open("/Users/tinaccil/Documents/GitHub/GreToBaPe_Cleaning/tmp_network_to
    output.write(tmp_endo.to_csv(sep="\t", index=False))
```

Convert pandas reaction network dataframe to graph raction network with species and reaction as different type of nodes

In [ ]:
```
net_g   = nx.DiGraph()
mol_vec = df_info_mol['species'].to_numpy()
mol_ene = (df_info_mol['Energy'] + df_info_mol['ZPE']).to_numpy()
net_id  = df_net_tmp['Number'].to_numpy()
net_rec = df_net_tmp['enthalpy'].to_numpy()
#create nodes and add attribute
for i,mol in enumerate(mol_vec):
    net_g.add_node(mol)
    tmp_attr = {'type': 'species','energy': mol_ene[i]}
    net_g.nodes[mol].update(tmp_attr.copy())
for i,mol in enumerate(ph_list):
    net_g.add_node(mol)
    tmp_attr = {'type': 'species','energy': np.nan}
    net_g.nodes[mol].update(tmp_attr.copy())
#create edges and attribute
for i,rec in enumerate(net):
    net_g.add_node(net_id[i])
    tmp_attr = {'type': 'reaction','enthalpy': net_rec[i]}
    net_g.nodes[net_id[i]].update(tmp_attr.copy())
    for j,item_j in enumerate(set(net[i].reactants)):
        if item_j not in nan:
            net_g.add_edge(item_j,net_id[i])
    for k,item_k in enumerate(set(net[i].products)):
        if item_k not in nan:
            net_g.add_edge(net_id[i],item_k)
```

DELETED ALL THE ENDOTHERMIC REACTIONS AND CONSEQUENT DOMINO EFFECT

it will generete a `_domino_endo.dat` file with all the reactions adresses as endothermic and deleted by Domino effect

also will create the clean network `_clean.dat`