HEADER

the python package and function in my lib to be upload.

- path_to_network --> path to the network to be clean

In [ ]:

```
import pandas as pd
import numpy as np
import networkx as nx
from lib import FromNetworkDataFrameToNetworkClassList, ExtractMolFromNetwork
path_to_network = '/Users/tinaccil/Documents/GitHub/GreToBaPe_Cleaning/tmp_ne
```

In [ ]:

```
#Encode KIDA .dat format into pandas dataframe
df_net = FromNetworkKidaDATtoCSV(path_to_network,save=False)
#Extract all the molecules inside the Network inside a dataframe and then to
df_mol = ExtractMolFromNetwork(df_net)
mol_vec  = df_mol['species'].to_numpy()
mol_prop = np.zeros(len(mol_vec),dtype=float)
#Add properties to species (EXAMPLE)
for i in range(len(mol_vec)):
    mol_prop[i] = len(mol_vec[i])
df_mol['properties'] = mol_prop
#Print network and species information
with pd.option_context('display.max_rows', None, 'display.max_columns', None)
    display(df_net)
with pd.option_context('display.max_rows', None, 'display.max_columns', None)
    display(df_mol)
#Network information and adding fake energy to Network (EXAMPLE)
net_id  = df_net['Number'].to_numpy()
net_rec = np.zeros(len(net_id),dtype=float)
net   = FromNetworkDataFrameToNetworkClassList(df_net)
#Define the network type
net_g   = nx.DiGraph()
#create nodes and add attribute
for i,mol in enumerate(mol_vec):
    net_g.add_node(mol)
    tmp_attr = {'type': 'species','energy': mol_prop[i]}
    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','energy': net_rec[i]}
    net_g.nodes[net_id[i]].update(tmp_attr.copy())
    for j,item_j in enumerate(set(net[i].reactants)):
        net_g.add_edge(item_j,net_id[i])
    for k,item_k in enumerate(set(net[i].products)):
        net_g.add_edge(net_id[i],item_k)
```

In [ ]:

```
#print Network
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 10))
pos = nx.spring_layout(net_g,weight=None)
plt.rcParams['font.family'] = 'Times New Roman'
edges = nx.draw_networkx_edges(net_g,
    pos,
    alpha=0.7,
    node_size=280,
    connectionstyle="arc3,rad=0.1",
    edge_color='black',
    width=2.5,
)
nodes = nx.draw_networkx_nodes(
        net_g,
        pos,
        nodelist=list(mol_vec),
        node_size=500,
        alpha = 0.5,
        node_color='blue',
        label=list(mol_vec),
    )
nx.draw_networkx_nodes(
    net_g,
    pos,
    nodelist=list(net_id),
    node_size=500,
    alpha = 0.5,
    node_color='red',
    node_shape='s',
)
nx.draw_networkx_labels(
    net_g,
    pos,
    font_size=14,
    font_weight="bold",
)
plt.tight_layout()
plt.show()
```