Reaction Topology Analysis

Formulation

In [1]:
```
import pandas as pd
from Utils.tools import *
import numpy as np
from IPython.display import display_html
import sympy
from scipy import stats
```

In [2]:
```
#Loading the dataframe.
path_to_csv = 'Utils\Datasets\Q4.csv'
rates_df = pd.read_csv(path_to_csv)
```

In [3]:
```
#A list of all metabolites present in the system.
metabolites = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I']
rxns_flux = rates_df['Unnamed: 0'].tolist()

#A list of all reactions. 0 represents system inflow and 1 represents system outflow.
topology_1 = [
    'v1: 1 -> A', 'v2: A -> C', 'v3: 1 -> B', 'v4: B -> D', 'v5: C -> D',
    'v6: C -> F', 'v7: C -> E', 'v8: D -> E', 'v9: D -> I', 'v10: E -> G',
    'v11: E -> H', 'v12: F -> G', 'v13: I -> H', 'v14: G -> -1', 'v15: H -> -1'
]

topology_2 = [
    'v1: 1 -> A', 'v7: A -> C', 'v3: 1 -> B', 'v4: B -> D', 'v5: C -> D',
    'v6: C -> F', 'v2: C -> E', 'v8: D -> E', 'v9: D -> I', 'v10: E -> G',
    'v11: E -> H', 'v12: F -> G', 'v13: I -> H', 'v14: G -> -1', 'v15: H -> -1'
]
```

In [4]:
```
def stoichiometric_matrix(topology,rxns_flux=rxns_flux):
    '''
    A function that converts the topology of a reaction network
    into a matrix.
    '''
    st_dict = {}
    #Iterating through all reactions.
    for directed_edge in topology:
        eq = directed_edge.replace(' ', '')
        rxn_rate, rxn = eq.split(':')
        reactant, product = rxn.split('->')
        #Here all stoichiometric coefficients are 1.
        coeff_dict = {}
        coeff_dict[reactant] = -1
        coeff_dict[product] = +1
        #Accounting for inflow and outflow of system.
        for key in ('1', '-1'):
            coeff_dict.pop(key, None)
        st_dict[rxn_rate] = coeff_dict

    st_df = pd.DataFrame(st_dict).fillna(value=0).astype(int)
    st_df = st_df.reindex(columns=rxns_flux)

    #Converting the dataframe into an numpy array.
    st_array = st_df.to_numpy()

    return st_array, st_df
```

Mass Balance Equations

In [5]:
```
def mass_balance_equations(topology: list):
    '''
    A function that prints the mass balance equations of a
    given topology.
    '''
    st_arr, st_df = stoichiometric_matrix(topology)
    #Creating mathematical symbols out of flux variables.
    flux_sym = [sympy.symbols(v) for v in st_df.columns]
    #Converting them in to a vector.
    flux_vect = sympy.Matrix(flux_sym)
    st_mat = sympy.Matrix(st_arr)
    #Mass balance equations are obtained by a dot product of stoichiometric matrix and flux.
    mass_bal_eqn = st_mat * flux_vect

    print('---' * 30)
    print('---' * 30)
    for metabolite, eqn in zip(st_df.index, range(mass_bal_eqn.shape[0])):
        print(
            f'Mass balance equation for {metabolite} is, \n\n \t\t\t\t{mass_bal_eqn[eqn]} = 0 \n '
        )
    print('---' * 30)
    print('---' * 30)
```

Topology 1 mass balance equations

In [6]:
```
mass_balance_equations(topology_1)
```

```
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
Mass balance equation for A is,

                                v1 - v2 = 0

Mass balance equation for C is,

                                v2 - v5 - v6 - v7 = 0

Mass balance equation for B is,

                                v3 - v4 = 0

Mass balance equation for D is,

                                v4 + v5 - v8 - v9 = 0

Mass balance equation for F is,

                                -v12 + v6 = 0

Mass balance equation for E is,

                                -v10 - v11 + v7 + v8 = 0

Mass balance equation for I is,

                                -v13 + v9 = 0

Mass balance equation for G is,

                                v10 + v12 - v14 = 0

Mass balance equation for H is,

                                v11 + v13 - v15 = 0

------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
```

Topology 2 mass balance equations

In [7]:
```
mass_balance_equations(topology_2)
```

```
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
Mass balance equation for A is,

                                v1 - v7 = 0

Mass balance equation for C is,

                                -v2 - v5 - v6 + v7 = 0

Mass balance equation for B is,

                                v3 - v4 = 0

Mass balance equation for D is,

                                v4 + v5 - v8 - v9 = 0

Mass balance equation for F is,

                                -v12 + v6 = 0

Mass balance equation for E is,

                                -v10 - v11 + v2 + v8 = 0

Mass balance equation for I is,

                                -v13 + v9 = 0

Mass balance equation for G is,

                                v10 + v12 - v14 = 0

Mass balance equation for H is,

                                v11 + v13 - v15 = 0

------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
```

Hypothesis Testing

- The null hypothesis for all the matabolites is their corresponding mass balance equation derived above and the alternate hypothesis is that the mass balance equations are not equal to zero.
- The statistical test to be employed is the t-test.
- Such a choice is justifiable because of the normal nature of the metabolites flux balance equations and the available number of samples being less than 30.
- The alpha value is given to be 0.05.

In [8]:
```
def metabolites_mass_bal(samples_df, topology, alpha=0.05):
    '''
    A function that calcuates the mass balance equations values for all
    the samples.

        Args:
              samples_df -> (dataframe) Measured reaction flux values.
              topology -> (list) The reaction network toplogy.
              alpha -> (float) confidence interval.
                       As default set to 0.05.
    '''

    #Estimating the mass balance eqution values.
    mass_bal_dict = {}

    st_mat, st_df = stoichiometric_matrix(topology)

    for sample in ['sample1', 'sample2', 'sample3']:
        mass_bal_dict[sample + '_mass_balance'] = np.dot(
            st_mat, samples_df[sample].values)

    df = pd.DataFrame(mass_bal_dict, index=st_df.index)

    #Running the t-test.
    test_results = stats.ttest_1samp(df, 0, axis=1)
    df['t-test statistic'] = test_results.statistic
    df['p-value'] = test_results.pvalue
    df['Null Hypothesis'] = (df['p-value'] > alpha).map({
        False: 'Rejected',
        True: 'Accepted'
    })

    return style_df(df)
```

Topology 1

In [9]:
```
metabolites_mass_bal(rates_df,topology_1)
```

Out[9]:

	sample1_mass_balance	sample2_mass_balance	sample3_mass_balance	t-test statistic	p-value	Null Hypothesis
A	0.455683	-1.757053	1.083929	-0.084124	0.940620	Accepted
C	1.264543	0.764740	-0.563927	0.895237	0.465132	Accepted
B	-1.065361	0.801704	1.496038	0.537120	0.644945	Accepted
D	0.931989	-0.621484	-3.551706	-0.821902	0.497524	Accepted
F	-0.433978	0.223297	-1.838179	-1.123319	0.378027	Accepted
E	-1.738409	-0.134649	0.206820	-0.926220	0.452112	Accepted
I	-4.401568	0.440471	0.315448	-0.762568	0.525385	Accepted
G	2.276287	-2.904792	1.786745	0.233777	0.836908	Accepted
H	5.291297	1.660844	0.732972	1.841850	0.206836	Accepted

Topology 2

In [10]:
```
metabolites_mass_bal(rates_df,topology_2)
```

Out[10]:

	sample1_mass_balance	sample2_mass_balance	sample3_mass_balance	t-test statistic	p-value	Null Hypothesis
A	50.856948	49.901162	51.270140	125.001206	0.000064	Rejected
C	-99.537987	-102.551689	-100.936349	-116.004316	0.000074	Rejected
B	-1.065361	0.801704	1.496038	0.537120	0.644945	Accepted
D	0.931989	-0.621484	-3.551706	-0.821902	0.497524	Accepted
F	-0.433978	0.223297	-1.838179	-1.123319	0.378027	Accepted
E	48.662856	51.523565	50.393031	60.339833	0.000275	Rejected
I	-4.401568	0.440471	0.315448	-0.762568	0.525385	Accepted
G	2.276287	-2.904792	1.786745	0.233777	0.836908	Accepted
H	5.291297	1.660844	0.732972	1.841850	0.206836	Accepted

- From the above two tables we can see that the null hypothesis is rejected for the metabolites A, C and E in the topology 2.
- This would mean that there would be some accumulation in this nodes.
- But such an occurence would contradict our assumptions of inflow flux being equal to output flux.
- Based on this we conclude that the **topology 1 is the actual reaction topology**.