Hypergeometric test to identify enrichment of housekeeping reactions in the proposed clutsters

Formualtion of the statistical test

The purpose of designing this statistical test is to identify whether the 26 clusters of reactions proposed by the scientist have an abundance of housekeeping reactio

To identify this, the following **Null Hypothesis** and **Alternate Hypothesis** are proposed.

$$H_0: \text{ The number of housekeeping reactions is as expected from a random sampling}$$

$$H_1: \text{ There is an enrichment of housekeeping reactions in the cluster}$$

Approach

- The alpha level is mentioned to be 0.05 and can be thought of as a critical point.
- Assuming that the null hypothesis is true (as one would expect from a random sampling that follows hypergeometric distribution), alpha value means that we reject this null hypothesis only if the observed data is so unusual that they would have occuered by chance at most 5 of the time.
- Given the sampling distribution to be hypergeometric we can estimate a statisitc associated with it called the p-value.This is an indicator of how extreme the data are.
- **Criterion**
  - p-value $\leq$ alpha, then we reject the null hypothesis and choose the alternate hypothesis.
  - p-value $>$ alpha, then we accept the null hypothesis.

Formulation to Code

In [1]:
```
from scipy.stats import hypergeom
#Importing style_df,csv_to_df,drop_nan.
from Utils.tools import *
```

In [2]:
```
def measure_enrichment(cluster: set, hk_set: set, num_rxns: int) -> dict:
    '''
    A function that returns the necessary values to make
    an inference about whether a given cluster has
    enrichment of housekeeping genes.
    '''

    enrich_dict = {}

    #Dropping the NaN elements if any.
    cluster = drop_nan(cluster)
    hk_set = drop_nan(hk_set)

    enrich_dict['Number of housekeeping reactions in the cluster'] = len(
        cluster & hk_set)
    enrich_dict['Total Number of reactions'] = num_rxns
    enrich_dict['Number of reactions in the cluster'] = len(cluster)
    enrich_dict['Total Number of housekeeping reactions'] = len(hk_set)

    #Parameters of the hypergeometric distribution.
    k, M, n, N = [*enrich_dict.values()]

    enrich_dict['p-value'] = hypergeom.sf(k, M, n, N)

    return enrich_dict


def analyse_clusters(clust_path,
                     hk_path,
                     analysing_function=measure_enrichment,
                     num_rxns=4551):
    '''
    A function that applies measure_enrichment on all the
    given clusters in the clust_df.
    '''
    clust_df, hk_df = csv_to_df([clust_path, hk_path])

    #Making a list of all clusters and then setting them as the index of the dataframe for easier analysis.
    clust_names = clust_df['Unnamed: 0'].tolist()
    clust_df.set_index('Unnamed: 0', inplace=True)

    #Converting the housekeeping reactions as a set.
    hk_set = set(hk_df['HK_reactions'])

    enrich_dict_list = []

    #Iterating through all the clusters.
    for clust_name in clust_names:

        cluster = clust_df.loc[clust_name]

        #Storing the enrich_dict of the cluster as a list.
        enrich_dict_list.append(measure_enrichment(cluster, hk_set, num_rxns))

    enrich_df = pd.DataFrame(enrich_dict_list)
    enrich_df.insert(loc=0, column='Cluster', value=clust_names)

    #Accepting or rejecting the null hypothesis.
    enrich_df['Null Hypothesis'] = (enrich_df['p-value'] > 0.05).map({
        False:
        'Rejected',
        True:
        'Accepted'
    })
    enrich_df['Alternate Hypothesis'] = enrich_df['Null Hypothesis'].map({
        'Accepted':
        'Rejected',
        'Rejected':
        'Accepted'
    })

    return style_df(enrich_df)
```

In [3]:
```
path_to_csv_files = {
    'clust_path': 'Utils/Datasets/Cluster_rxn_set.csv',
    'hk_path': 'Utils/Datasets/HK_rxns.csv'
}
analyse_clusters(**path_to_csv_files)
```

Out[3]:

	Cluster	Number of housekeeping reactions in the cluster	Total Number of reactions	Number of reactions in the cluster	Total Number of housekeeping reactions	p-value	Null Hypothesis	Alternate Hypothesis
0	cluster1	650	4551	1888	929	0.000000	Rejected	Accepted
1	cluster2	220	4551	792	929	0.000000	Rejected	Accepted
2	cluster3	16	4551	223	929	1.000000	Accepted	Rejected
3	cluster4	37	4551	191	929	0.601612	Accepted	Rejected
4	cluster5	54	4551	439	929	0.999998	Accepted	Rejected
5	cluster6	41	4551	408	929	1.000000	Accepted	Rejected
6	cluster7	17	4551	191	929	0.999991	Accepted	Rejected
7	cluster8	26	4551	269	929	0.999999	Accepted	Rejected
8	cluster9	62	4551	273	929	0.147352	Accepted	Rejected
9	cluster10	18	4551	167	929	0.999412	Accepted	Rejected
10	cluster11	134	4551	491	929	0.000040	Rejected	Accepted
11	cluster12	18	4551	112	929	0.850379	Accepted	Rejected
12	cluster13	1	4551	49	929	0.999822	Accepted	Rejected
13	cluster14	48	4551	242	929	0.552984	Accepted	Rejected
14	cluster15	4	4551	119	929	1.000000	Accepted	Rejected
15	cluster16	5	4551	65	929	0.995558	Accepted	Rejected
16	cluster17	15	4551	124	929	0.989849	Accepted	Rejected
17	cluster18	0	4551	3	929	0.495976	Accepted	Rejected
18	cluster19	0	4551	1	929	0.204131	Accepted	Rejected
19	cluster20	20	4551	98	929	0.440783	Accepted	Rejected
20	cluster21	95	4551	429	929	0.159137	Accepted	Rejected
21	cluster22	2	4551	3	929	0.008484	Rejected	Accepted
22	cluster23	28	4551	47	929	0.000000	Rejected	Accepted
23	cluster24	6	4551	17	929	0.041367	Rejected	Accepted
24	cluster25	8	4551	13	929	0.000191	Rejected	Accepted
25	cluster26	0	4551	106	929	1.000000	Accepted	Rejected