**##### Load the libraries #####**

**import** numpy **as** np

**import** pandas **as** pd

**import** matplotlib.pyplot **as** plt

**import** random

**from** sklearn.neural\_network **import** MLPRegressor

**from** sklearn.model\_selection **import** train\_test\_split

**#####Load the wildtype dataset ####**

df **=** pd**.**read\_csv('wildtype\_dataset.csv', sep**=**',')

**######Split the dataset**

train, test **=** train\_test\_split(df, test\_size**=**0.2,random\_state**=**10)

**##### Autoencoder modelling (wildtype trained) ###**

*# Shape of input and latent variable*

n\_input **=** 821 *# No. of neurons in the input layer, based on the dimension of feature #*

*# Encoder structure*

n\_encoder1 **=** 600

n\_encoder2 **=** 300

n\_encoder3 **=** 150

n\_latent **=** 75 *## Reduced dimension (neurons) of the latent space ##*

*# Decoder structure*

n\_decoder3 **=** 150

n\_decoder2 **=** 300

n\_decoder1 **=** 600

*## Model ##*

reg **=** MLPRegressor(hidden\_layer\_sizes **=** (n\_encoder1, n\_encoder2, n\_encoder3, n\_latent, n\_decoder3, n\_decoder2, n\_decoder1),

activation **=** 'relu',

solver **=** 'adam',

learning\_rate\_init **=** 0.0001,

max\_iter **=** 200,

tol **=** 0.0000001, random\_state**=**10,

verbose **=** **True**)

***## Train the model ##***

reg**.**fit(train, train)

***#### Predict wildtype test dataset using wildtype dataset trained autoencoder model ###***

test\_reconst **=** reg**.**predict(test)

***#Compute DIO (wildtpe trained, wildtype test)***

dio **=** test **-** test\_reconst

*# saving the dataframe*

dio**.**to\_csv('dio\_wildtest.csv', header**=True**, sep**=**',', index**=None**)

**#####DIO Plotting**

npdio**=**dio**.**to\_numpy()

dfdio **=** npdio**.**flatten()

plt**.**hist(dfdio, weights**=**np**.**ones\_like(dfdio) **/** len(dfdio), bins**=**np**.**arange(**-**10.5, 10.5), alpha**=**0.9, rwidth**=**0.5)

plt**.**savefig("wildtest\_dio.png")

**#####Load the wildtype dataset ####**

df1 **=** pd**.**read\_csv('mutantype\_dataset.csv', sep**=**',')

**######Split the dataset**

train1, test1 **=** train\_test\_split(df1, test\_size**=**0.2,random\_state**=**10)

**###Predict mutant test dataset using wildtype trained autoencoder model #####**

test1\_reconst **=** reg**.**predict(test1)

***# Compute DIO (wildtpe trained, wildtype test)***

dio1 **=** test1 **-** test1\_reconst

#####save the Dataframe

dio1**.**to\_csv('dio\_mutanttest.csv', header**=True**, sep**=**',', index**=None**)

**### DIO Plotting ###**

npdio1**=**dio1**.**to\_numpy()

dfdio1 **=** npdio1**.**flatten()

plt**.**hist(dfdio1, weights**=**np**.**ones\_like(dfdio1) **/** len(dfdio1), bins**=**np**.**arange(**-**10.5, 10.5), alpha**=**0.9, rwidth**=**0.5)

plt**.**savefig("mutanttest\_dio.png")