

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")
```