In [1]:

```
#The following document contains code to run IF-PCA+ one time.
```

In [2]:

```
import pandas as pd
import numpy as np
import scipy
import scipy.io
from scipy.linalg import svd
from sklearn.cluster import KMeans
from IFPCA import get_IFPCA_acc, get_acc_PCA
```

In [ ]:

In [3]:

```
########### Load denoised Single-Cell data  ###########
### Name set: (load p-by-n data matrix)
#camp1, 13111-by-777
#camp2, 11233-by-734
#darmanis, 13400-by-466
#deng, 16347-by-268
#goolam, 21199-by-124
#grun, 5547-by-1502

#li, 25369-by-561
#patel, 5948-by-430

path = "Dataset/sc_data/" # path of the data sets
name = 'grun'

# First load original dataset
Data = pd.read_csv(path + name + '-x-filter.txt', sep=",", header = None) # p
y = pd.read_csv(path + name + '-y.txt', sep=" ", header=None).iloc[:,0] # n-b
K = max(y) - min(y) + 1 #number of classes
X = Data.T
X= X.to_numpy()
n, param_size = np.shape(X)
print("num of clusters:",K)
print("data shape:",np.shape(X))


# Then load denoise data from the MAT-file
mani_file = name +  '_Y2_denoised_data.mat'
mani_data = scipy.io.loadmat(mani_file)
log_x = np.array(mani_data['fea_log_fit'])
log_duo_x = np.array(mani_data['fea_log_duo_fit'])
```

```
num of clusters: 2
data shape: (1502, 5547)
```

In [4]:

```
# ########### Load denoised Micro array data  ###########
# ### Name set: (load p-by-n data matrix)
# # brain, 5597-by-42
# # breast, 22215-by-276
# # colon, 2000-by-62
# # leukemia, 3571-by-72
# # lung1, 12533-by-181
# # lung2, 12600-by-203
# # lymphoma, 4026-by-62
# # prostate, 6033-by-102
# # srbct, 2308-by-63
# # su, 7909-by-174

# path = "Dataset/microarray_data/" # path of the data sets
# name = 'colon'

# # First load original dataset
# Data = pd.read_csv(path + name + '.x.txt', sep=",", header = None) # p-by-n
# y = pd.read_csv(path + name + '.y.txt', sep=" ", header=None).T.values[0]
# K = max(y) - min(y) + 1 #number of classes
# X = Data.T
# X= X.to_numpy()
# n, param_size = np.shape(X)
# print("num of clusters:",K)
# print("data shape:",np.shape(X))


# # Then load denoise data from the MAT-file
# mani_file = name +  '_Y2_denoised_data.mat'
# mani_data = scipy.io.loadmat(mani_file)
# ori_x = np.array(mani_data['fea_raw_fit'])
# ori_duo_x = np.array(mani_data['fea_raw_duo_fit'])
```

Run IF-PCA and IF-PCA+ one time

In [5]:

```python
import time

# pca_acc    = get_acc_PCA(X, y, K)
# IF_PCA_acc = get_IFPCA_acc(X, y, K)

start_time = time.time()
IF_PCA_Plus_acc = get_IFPCA_acc(log_duo_x, y, K,manifold = True)

end_time = time.time()

runtime = end_time - start_time
print(f"Runtime: {runtime} seconds")
```

```
Running IF steps
Running PCA (for manifold)

c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, whe
n there are less chunks than available threads. You can avoid it by setting
the environment variable OMP_NUM_THREADS=6.
  warnings.warn(
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, whe
n there are less chunks than available threads. You can avoid it by setting
the environment variable OMP_NUM_THREADS=6.
  warnings.warn(

[0.9327563249001332, 0.9327563249001332, 0.9327563249001332, 0.932756324900
1332, 0.9327563249001332]
Runtime: 15.582125186920166 seconds
```

```
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, whe
n there are less chunks than available threads. You can avoid it by setting
the environment variable OMP_NUM_THREADS=6.
  warnings.warn(
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, whe
n there are less chunks than available threads. You can avoid it by setting
the environment variable OMP_NUM_THREADS=6.
  warnings.warn(
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1412:
FutureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
c:\Users\Yi G\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1436:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, whe
n there are less chunks than available threads. You can avoid it by setting
the environment variable OMP_NUM_THREADS=6.
  warnings.warn(
```

In [6]:

```
#### Print single cell results
print ("Single cell dataset", name)
# print("pca", pca_acc)
# print("IF-PCA", IF_PCA_acc)
print("IF-PCA plus", IF_PCA_Plus_acc)
```

```
Single cell dataset grun
IF-PCA plus 0.9327563249001332
```

In [ ]: