```
###import packages
import numpy as np
import pandas as pd
import sys
import os
import time as tm
import pickle
from functools import partial
import scipy.stats as st
from scipy.stats import wasserstein_distance
import scipy.stats
import copy
from sklearn.model_selection import KFold
import pandas as pd
import multiprocessing
import matplotlib as mpl
import matplotlib.pyplot as plt
import scanpy as sc
import warnings
import subprocess
import seaborn as sns
from sklearn.metrics import mean_squared_error
from scipy.spatial.distance import jensenshannon
from scipy.stats import pearsonr,ttest_ind,mannwhitneyu
import matplotlib

import Benchmarking.CellAssigment as CellAssigment
import Benchmarking.util as util




## First, you must download the data and put it in "Rawdata" directory
## The scRNA count link: https://drive.google.com/file/d/
    1ugoZtGHwoK6CLGVK4kWehLscqJqNNha4/view?usp=sharing


## Predict gene spatial distribution of undetected genes

# You can import the package "SpatialGenes" to directly predict the gene
    spatial distribution for any spatial datasets.

# Before run pipeline, please prepare the following files:
#1): scRNA count files;
#2): spatial count files;
#3): spatial location files for novoSpaRc and SpaOTsc;
#4): count files containing the number of cells in each space point for
    Tangram(option).

# For more details, please see the Benchmarking/SpatialGenes.py and Figure
    Data

import os
import numpy as np
import pandas as pd
import Benchmarking.SpatialGenes as SpatialGenes
import os

PATH = 'FigureData/Figure2/Dataset2_osmFISH/Rawdata/'
```

```
RNA_path = PATH + 'scRNA_count.txt'
Spatial_path =  PATH + 'Insitu_count.txt'
location_path = PATH + 'Locations.txt'
RNA_data = pd.read_table(RNA_path,header=0,index_col = 0)
Spatial_data = pd.read_table(Spatial_path,sep='\t',header=0)
train_list = list(RNA_data.index&Spatial_data.columns)
print (train_list)
test_list = list(set(RNA_data.index) - set(Spatial_data.columns))[:20]

outdir =  'FigureData/Figure2/Dataset2_osmFISH/Test/'
if not os.path.exists(outdir):
        os.mkdir(outdir)
test = SpatialGenes.GenePrediction(RNA_path, Spatial_path, location_path,
    train_list = train_list, test_list = test_list, outdir = outdir)
Methods =
    ['SpaGE','novoSpaRc','Tangram','gimVI','Tangram_image','Seurat','LIGER']
Result = test.Imputing(Methods)


### Prediction Cell Locations

#You can import the package "CellAssigment" to directly predict the cell
    locations for any spatial datasets.

# Before run pipeline, please prepare the following files:
#1): scRNA count files;
#2): spatial count files;
# 3): scRNA cell annotation files;
#4): count files containing the number of cell types in each space point
    (option).

# For more details, please see the Benchmarking/CellAssigment.py and Figure
    Data


PATH = 'FigureData/Figure4/Dataset7_STARmap/'

scRNA = PATH + 'Rawdata/scRNA_count.txt'
spatial_count = PATH + 'Simulated_STARmap/combined_spatial_count.txt'
cell_counts = PATH + 'Simulated_STARmap/combined_cell_counts.txt'
scrna_meta = PATH + 'Rawdata/scRNA_annotate.txt'
annotatetype = 'subclass'
gd_result = PATH + 'Simulated_STARmap/combined_spot_clusters.txt'
outdir = PATH + 'Result_STARmap/'
if not os.path.exists(outdir):
    os.mkdir(outdir)
location = PATH + 'Simulated_STARmap/combined_Locations.txt'



MC = CellAssigment.MappingCell(RNA_path = scRNA, Spatial_path = spatial_count,
    location_path = location,
                               count_path = cell_counts, scrna_annotation =
                                   scrna_meta, gd_result = gd_result,
                               annotatetype = annotatetype, outdir = outdir)
Tools = ['novoSpaRc','Seurat','SpaOTsc','Tangram']
Tools = ['novoSpaRc','SpaOTsc','Tangram']
MC.workstart(Tools)
```

```
### Calculate the accuracy of prediction

PATH = 'FigureData/Figure4/Dataset7_STARmap/'
Methods = ['Seurat','Tangram','SpaOTsc','novoSpaRc']
outdir = PATH + 'Result_STARmap/'
util.CalculateMetric(outdir, Methods, PATH + 'Simulated_STARmap/
    combined_spot_clusters.txt')
```