Tutorial

January 4, 2022

[1]:
```
##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 time
```

[2]:
```
###please in the SpatialBenchmarking dir.
os.chdir('../')
```

[ ]:
```
##First, you must download the data
## The data 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 forecasting, please prepare the following files:
#1): scRNA count files;
#2): spatial count files;
#3): spatial location files for novoSpaRc and SpaOTsc;

# 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 = '../DataUpload/Dataset15/'
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 =  'Dataset15/'
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', 'Seurat', 'SpaOTsc', 'novoSpaRc', 'LIGER', 'stPlus']
Result = test.Imputing(Methods)
```

1 GPU Platform gimVI, Tangram, and stPlus

[ ]:
```
import os
import numpy as np
import pandas as pd
import Benchmarking.SpatialGenes as SpatialGenes
import os
from stPlus import *

PATH = '../DataUpload/Dataset15/'
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 =  'Dataset15/'
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 = ['Tangram', 'gimVI', 'stPlus']
Result = test.Imputing(Methods)
```

[ ]:
```
#You can import the package "DeconvolutionSpot" to directly predict the cell␣
 ↪locations for any spatial datasets.

# Before forecasting, please prepare the following files:
#1): scRNA count files, h5ad file or h5seurat file;
#2): spatial count files, h5ad file or h5seurat file;
# 3): scRNA cell annotation files;
#4): output dir.

# For more details, please see the Benchmarking/DeconvolutionSpot.py and Figure␣
 ↪Data
```

2 Prediction Celltype deconvolution

[ ]:
```
import Benchmarking.DeconvolutionSpot as DeconvolutionSpot
RNA_file = '../Decon/Dataset10_STARmap/starmap_sc_rna.tsv'
RNA_h5ad = '../Decon/Dataset10_STARmap/starmap_sc_rna.h5ad'
RNA_h5Seurat = '../Decon/Dataset10_STARmap/starmap_sc_rna.h5seurat'

Spatial_file = '../Decon/Dataset10_STARmap/starmap_spatial.tsv'
Spatial_h5ad = '../Decon/Dataset10_STARmap/starmap_spatial.h5ad'
Spatial_h5Seurat = '../Decon/Dataset10_STARmap/starmap_spatial.h5seurat'

celltype_key = 'celltype'
celltype_file = '../Decon/Dataset10_STARmap/starmap_sc_rna_celltype.tsv'

output_path = 'FigureData/Figure4/Dataset4_seqFISH+/'
if not os.path.exists(output_path):
    os.mkdir(output_path)

test = DeconvolutionSpot.Deconvolutions(RNA_file, RNA_h5ad, RNA_h5Seurat,␣
 ↪Spatial_file, Spatial_h5ad, Spatial_h5Seurat, celltype_key, celltype_file,␣
 ↪output_path)
```

```
Methods =␣
 ↪['Cell2location','SpatialDWLS','RCTD','STRIDE','Stereoscope','Tangram','DestVI',␣
 ↪'Seurat', 'SPOTlight', 'DSTG']
Result = test.Dencon(Methods)
```

[ ]: