comprehensive+Liang

June 26, 2022

1 importing pandas package

```
[1]: import pandas as pd
```

2 Reading the ATG genes (comprehesive list) databse

```
[2]: comprehensive = pd.read_excel('comprehensive + dark genes.
      ↪xlsx',engine='openpyxl')
```

3 Reading the AD affected in Liang's databse and extracting the list of genes at different sheets

```
[4]: AD_affected_EC=pd.read_excel('AD affected.
      ↪xlsx',sheet_name='EC',engine='openpyxl')
     AD_affected_HIP=pd.read_excel('AD affected.
      ↪xlsx',sheet_name='HIP',engine='openpyxl')
     AD_affected_PC=pd.read_excel('AD affected.
      ↪xlsx',sheet_name='PC',engine='openpyxl')
     AD_affected_MTG=pd.read_excel('AD affected.
      ↪xlsx',sheet_name='MTG',engine='openpyxl')
     AD_affected_SFG=pd.read_excel('AD affected.
      ↪xlsx',sheet_name='SFG',engine='openpyxl')
     AD_affected_VCX=pd.read_excel('AD affected.
      ↪xlsx',sheet_name='VCX',engine='openpyxl')
     AD_affected_EC=list(AD_affected_EC['symbol'])
     AD_affected_HIP=list(AD_affected_HIP['symbol'])
     AD_affected_PC=list(AD_affected_PC['symbol'])
     AD_affected_MTG=list(AD_affected_MTG['symbol'])
     AD_affected_SFG=list(AD_affected_SFG['symbol'])
     AD_affected_VCX=list(AD_affected_VCX['symbol'])
```

4 Filtering "Liang (AD-Affected)" in selecting rows with having the same gene with comprehensive list in a "symbol" column.

[14]:
```
intresect_comprehensive_AD_affected_EC = comprehensive[comprehensive['symbol'].
 ↪isin(AD_affected_EC)]
intresect_comprehensive_AD_affected_HIP = comprehensive[comprehensive['symbol'].
 ↪isin(AD_affected_HIP)]
intresect_comprehensive_AD_affected_PC = comprehensive[comprehensive['symbol'].
 ↪isin(AD_affected_PC)]
intresect_comprehensive_AD_affected_MTG = comprehensive[comprehensive['symbol'].
 ↪isin(AD_affected_MTG)]
intresect_comprehensive_AD_affected_SFG= comprehensive[comprehensive['symbol'].
 ↪isin(AD_affected_SFG)]
intresect_comprehensive_AD_affected_VCX= comprehensive[comprehensive['symbol'].
 ↪isin(AD_affected_VCX)]
```

5 Saving the filtering Liang(AD affected) based on comprehensive list

[21]:
```
writer = pd.ExcelWriter('AD-Affected+comprehnsive.xlsx', engine='xlsxwriter')
intresect_comprehensive_AD_affected_EC.to_excel(writer, 'EC')
intresect_comprehensive_AD_affected_HIP.to_excel(writer, 'HIP')
intresect_comprehensive_AD_affected_PC.to_excel(writer, 'PC')
intresect_comprehensive_AD_affected_MTG.to_excel(writer, 'MTG')
intresect_comprehensive_AD_affected_SFG.to_excel(writer,'SFG')
intresect_comprehensive_AD_affected_VCX.to_excel(writer,'VCX')
writer.save()
```

6 Reading the Non demented in Liang's databse and extracting the list of genes at different sheets

[22]:
```
Non_demented_EC=pd.read_excel('Liang-non-demented.xlsx',sheet_name='entorhinal␣
 ↪cortex',engine='openpyxl')
Non_demented_HIP=pd.read_excel('Liang-non-demented.
 ↪xlsx',sheet_name='hippocampus',engine='openpyxl')
Non_demented_PC=pd.read_excel('Liang-non-demented.xlsx',sheet_name='middle␣
 ↪temporal gyrus',engine='openpyxl')
Non_demented_MTG=pd.read_excel('Liang-non-demented.xlsx',sheet_name='posterior␣
 ↪cingulate corrtex',engine='openpyxl')
Non_demented_SFG=pd.read_excel('Liang-non-demented.xlsx',sheet_name='superior␣
 ↪frontal gyrus',engine='openpyxl')
Non_demented_VCX=pd.read_excel('Liang-non-demented.xlsx',sheet_name='primary␣
 ↪visual cortex',engine='openpyxl')
```

```
Non_demented_EC=list(Non_demented_EC['symbol'])
Non_demented_HIP=list(Non_demented_HIP['symbol'])
Non_demented_PC=list(Non_demented_PC['symbol'])
Non_demented_MTG=list(Non_demented_MTG['symbol'])
Non_demented_SFG=list(Non_demented_SFG['symbol'])
Non_demented_VCX=list(Non_demented_VCX['symbol'])
```

7 Filtering “Liang (Non demented)” in selecting rows with having the same genes with comprehensive list in a “symbol” column.

[23]:

```
intresect_comprehensive_Non_demented_EC = comprehensive[comprehensive['symbol'].
 ↪isin(Non_demented_EC)]
intresect_comprehensive_Non_demented_HIP =␣
 ↪comprehensive[comprehensive['symbol'].isin(Non_demented_HIP)]
intresect_comprehensive_Non_demented_PC = comprehensive[comprehensive['symbol'].
 ↪isin(Non_demented_PC)]
intresect_comprehensive_Non_demented_MTG =␣
 ↪comprehensive[comprehensive['symbol'].isin(Non_demented_MTG)]
intresect_comprehensive_Non_demented_SFG= comprehensive[comprehensive['symbol'].
 ↪isin(Non_demented_SFG)]
intresect_comprehensive_Non_demented_VCX= comprehensive[comprehensive['symbol'].
 ↪isin(Non_demented_VCX)]
```

8 Saving the filtering Liang(Non demented) based on comprehensive list

[31]:

```
writer = pd.ExcelWriter('Non-demented+comprehnsive.xlsx', engine='xlsxwriter')
intresect_comprehensive_Non_demented_EC.to_excel(writer, 'EC')
intresect_comprehensive_Non_demented_HIP.to_excel(writer, 'HIP')
intresect_comprehensive_Non_demented_MTG.to_excel(writer, 'MTG')
intresect_comprehensive_Non_demented_PC.to_excel(writer, 'PC')
intresect_comprehensive_Non_demented_SFG.to_excel(writer, 'SFG')
intresect_comprehensive_Non_demented_VCX.to_excel(writer, 'VCX')
writer.save()
```

9 Reading the Normal aged in Liang’s databse and extracting the list of genes at different sheets

[7]:

```
Normal_aged_EC=pd.read_excel('Liang-normal-aged.
 ↪xlsx',sheet_name='EC',engine='openpyxl')
Normal_aged_HIP=pd.read_excel('Liang-normal-aged.
 ↪xlsx',sheet_name='HIP',engine='openpyxl')
```

```
Normal_aged_MTG=pd.read_excel('Liang-normal-aged.
 ↪xlsx',sheet_name='MTG',engine='openpyxl')
Normal_aged_PC=pd.read_excel('Liang-normal-aged.
 ↪xlsx',sheet_name='PC',engine='openpyxl')
Normal_aged_SFG=pd.read_excel('Liang-normal-aged.
 ↪xlsx',sheet_name='SFG',engine='openpyxl')
Normal_aged_VCX=pd.read_excel('Liang-normal-aged.
 ↪xlsx',sheet_name='VCX',engine='openpyxl')
Normal_aged_EC=list(Normal_aged_EC['symbol'])
Normal_aged_HIP=list(Normal_aged_HIP['symbol'])
Normal_aged_PC=list(Normal_aged_PC['symbol'])
Normal_aged_MTG=list(Normal_aged_MTG['symbol'])
Normal_aged_SFG=list(Normal_aged_SFG['symbol'])
Normal_aged_VCX=list(Normal_aged_VCX['symbol'])
```

10 Filtering “Liang (Normal aged)” in selecting rows with having the same genes with comprehensive list in a “symbol” column.

[8]:

```
intresect_comprehensive_Normal_aged_EC = comprehensive[comprehensive['symbol'].
 ↪isin(Normal_aged_EC)]
intresect_comprehensive_Normal_aged_HIP = comprehensive[comprehensive['symbol'].
 ↪isin(Normal_aged_HIP)]
intresect_comprehensive_Normal_aged_PC = comprehensive[comprehensive['symbol'].
 ↪isin(Normal_aged_PC)]
intresect_comprehensive_Normal_aged_MTG = comprehensive[comprehensive['symbol'].
 ↪isin(Normal_aged_MTG)]
intresect_comprehensive_Normal_aged_SFG= comprehensive[comprehensive['symbol'].
 ↪isin(Normal_aged_SFG)]
intresect_comprehensive_Normal_aged_VCX= comprehensive[comprehensive['symbol'].
 ↪isin(Normal_aged_VCX)]
```

11 Saving the filtering Liang(Normal aged) based on comprehensive list

[44]:

```
writer = pd.ExcelWriter('Normal-aged+comprehensive.xlsx', engine='xlsxwriter')
intresect_comprehensive_Normal_aged_MTG.to_excel(writer, 'MTG')
intresect_comprehensive_Normal_aged_HIP.to_excel(writer, 'HIP')
intresect_comprehensive_Normal_aged_EC.to_excel(writer, 'EC')
intresect_comprehensive_Normal_aged_PC.to_excel(writer, 'EC')
writer.save()
```

[ ]:

```

```