main

July 12, 2021

1 Tissue comparison for differential expression analysis

[1]:

```python
import functools
import numpy as np
import pandas as pd
from gtfparse import read_gtf
```

1.1 Configuration dictionary

[2]:

```python
config = {
    'caudate': '../../../caudate/_m/genes/diffExpr_EAvsAA_full.txt',
    'dlpfc': '../../../dlpfc/_m/genes/diffExpr_EAvsAA_full.txt',
    'hippo': '../../../hippocampus/_m/genes/diffExpr_EAvsAA_full.txt',
    'gyrus': '../../../dentateGyrus/_m/genes/diffExpr_EAvsAA_full.txt'
}
```

1.2 Functions

1.2.1 Cached functions

[3]:

```python
@functools.lru_cache()
def get_gtf(gtf_file):
    return read_gtf(gtf_file)


@functools.lru_cache()
def get_deg(filename):
    dft = pd.read_csv(filename, sep='\t', index_col=0)
    dft['Feature'] = dft.index
    dft['Dir'] = np.sign(dft['t'])
    if 'gene_id' in dft.columns:
        dft['ensemblID'] = dft.gene_id.str.replace('\\..*', '', regex=True)
    elif 'ensembl_gene_id' in dft.columns:
        dft.rename(columns={'ensembl_gene_id': 'ensemblID'}, inplace=True)
    return dft[['Feature', 'ensemblID', 'adj.P.Val', 'logFC', 't', 'Dir']]

@functools.lru_cache()
def get_deg_sig(filename):
    dft = get_deg(filename)
    return dft[(dft['adj.P.Val'] < 0.05)]


@functools.lru_cache()
def merge_dataframes(tissue1, tissue2):
    return get_deg(config[tissue1]).merge(get_deg(config[tissue2]),
                                          on='Feature',
                                          suffixes=['_%s' % tissue1, '_%s' %␣
 ↪tissue2])


@functools.lru_cache()
def merge_dataframes_sig(tissue1, tissue2):
    return get_deg_sig(config[tissue1]).merge(get_deg_sig(config[tissue2]),
                                              on='Feature',
                                              suffixes=['_%s' % tissue1, '_%s'␣
 ↪% tissue2])
```

1.2.2 Simple functions

[4]:

```
def tissue_annotation(tissue):
    return {'dlpfc': 'DLPFC', 'hippo': 'Hippocampus',
            'caudate': 'Caudate', 'gyrus': 'Dentate Gyrus'}[tissue]


def save_plot(p, fn, width=7, height=7):
    '''Save plot as svg, png, and pdf with specific label and dimension.'''
    for ext in ['.svg', '.png', '.pdf']:
        p.save(fn+ext, width=width, height=height)


def gene_annotation(gtf_file, feature):
    gtf0 = get_gtf(gtf_file)
    gtf = gtf0[gtf0["feature"] == feature]
    return gtf[["gene_id", "gene_name", "transcript_id", "exon_id",
                "gene_type", "seqname", "start", "end", "strand"]]
```

1.3 Gene annotation

[5]:

```
gtf_file = '/ceph/genome/human/gencode25/gtf.CHR/_m/gencode.v25.annotation.gtf'
gtf_annot = gene_annotation(gtf_file, 'gene')
gtf_annot.head(2)
```

```
INFO:root:Extracted GTF attributes: ['gene_id', 'gene_type', 'gene_status',
'gene_name', 'level', 'havana_gene', 'transcript_id', 'transcript_type',
'transcript_status', 'transcript_name', 'transcript_support_level', 'tag',
'havana_transcript', 'exon_number', 'exon_id', 'ont', 'protein_id', 'ccdsid']
```

```
[5]:               gene_id gene_name transcript_id exon_id  \
    0   ENSG00000223972.5   DDX11L1
    12  ENSG00000227232.5    WASH7P

                                   gene_type seqname  start    end strand
    0   transcribed_unprocessed_pseudogene    chr1  11869  14409      +
    12              unprocessed_pseudogene    chr1  14404  29570      -
```

1.4 BrainSeq Comparison

1.4.1 Summary of DE results

```
[6]: caudate = get_deg(config['caudate'])
     caudate.groupby('Dir').size()
```

```
[6]: Dir
     -1.0    10767
      1.0    11607
     dtype: int64
```

```
[7]: caudate[(caudate['adj.P.Val'] < 0.05)].shape
```

```
[7]: (2970, 6)
```

```
[8]: dlpfc = get_deg(config['dlpfc'])
     dlpfc.groupby('Dir').size()
```

```
[8]: Dir
     -1.0    11691
      1.0    10707
     dtype: int64
```

```
[9]: dlpfc[(dlpfc['adj.P.Val'] < 0.05)].shape
```

```
[9]: (2760, 6)
```

```
[10]: hippo = get_deg(config['hippo'])
      hippo.groupby('Dir').size()
```

```
[10]: Dir
      -1.0    11213
       1.0    11056
      dtype: int64
```

```
[11]: hippo[(hippo['adj.P.Val'] < 0.05)].shape
```

```
[11]: (2956, 6)
```

[12]:
```
gyrus = get_deg(config['gyrus'])
gyrus.groupby('Dir').size()
```

[12]:
```
Dir
-1.0    10855
 1.0    10285
dtype: int64
```

[13]:
```
gyrus[(gyrus['adj.P.Val'] < 0.05)].shape
```

[13]:
```
(786, 6)
```

1.4.2 Upset Plot

[14]:
```
phase2_dlpfc = dlpfc[(dlpfc['adj.P.Val'] < 0.05)].copy()
phase2_dlpfc['DLPFC'] = 1
phase2_dlpfc = phase2_dlpfc[['ensemblID', 'DLPFC']]

phase2_hippo = hippo[(hippo['adj.P.Val'] < 0.05)].copy()
phase2_hippo['Hippocampus'] = 1
phase2_hippo = phase2_hippo[['ensemblID', 'Hippocampus']]

phase3_caudate = caudate[(caudate['adj.P.Val'] < 0.05)].copy()
phase3_caudate['Caudate'] = 1
phase3_caudate = phase3_caudate[['ensemblID', 'Caudate']]

dentate_gyrus = gyrus[(gyrus['adj.P.Val'] < 0.05)].copy()
dentate_gyrus['Dentate Gyrus'] = 1
dentate_gyrus = dentate_gyrus[['ensemblID', 'Dentate Gyrus']]
```

[15]:
```
geneList = pd.merge(phase3_caudate[['ensemblID']],
                    phase2_dlpfc[['ensemblID']],
                    on=['ensemblID'], how='outer')\
             .merge(phase2_hippo[['ensemblID']],
                    on=['ensemblID'], how='outer')\
             .merge(dentate_gyrus[['ensemblID']],
                    on=['ensemblID'], how='outer')\
             .groupby(['ensemblID']).first().reset_index()

newC = pd.merge(geneList, phase3_caudate, on=['ensemblID'],
                how='outer').fillna(0)
newC['Caudate'] = newC['Caudate'].astype('int')

newD1 = pd.merge(geneList, phase2_dlpfc, on=['ensemblID'],
                 how='outer').fillna(0)
newD1['DLPFC'] = newD1['DLPFC'].astype('int')
```

```
newH = pd.merge(geneList, phase2_hippo, on=['ensemblID'],
                how='outer').fillna(0)
newH['Hippocampus'] = newH['Hippocampus'].astype('int')

newG = pd.merge(geneList, dentate_gyrus, on=['ensemblID'],
                how='outer').fillna(0)
newG['Dentate Gyrus'] = newG['Dentate Gyrus'].astype('int')

print(newC.shape, newH.shape, newD1.shape, newG.shape)
```

```
(6259, 2) (6259, 2) (6259, 2) (6259, 2)
```

[16]:

```
df = pd.concat([newC.set_index(['ensemblID']),
                newD1.set_index(['ensemblID']),
                newH.set_index(['ensemblID']),
                newG.set_index(['ensemblID'])],
               axis=1, join='outer')
df.head(2)
```

[16]:

```
                 Caudate  DLPFC  Hippocampus  Dentate Gyrus
ensemblID
ENSG00000001084        0      0            1              1
ENSG00000001460        0      1            1              0
```

[17]:

```
%load_ext rpy2.ipython
```

[18]:

```
%%R
#library(UpSetR)
#upset(df, order.by="freq", text.scale=c(3, 2.5, 2.4, 2.25, 2.6, 2.6), point.
 ↪size=3.6, line.size=1.4)
library(ComplexHeatmap)
subset_pvalue <- function(filename, fdr_cutoff){
    df <- subset(read.delim(filename, row.names=1, stringsAsFactors = F),
                 adj.P.Val < fdr_cutoff)
    if('gene_id' %in% colnames(df)){
        df$ensemblID <- gsub('\\..*', '', df$gene_id)
    } else if('ensembl_gene_id' %in% colnames(df)){
        df <- dplyr::rename(df, ensemblID=ensembl_gene_id)
    }
    return(df$ensemblID)
}

caudate = subset_pvalue('../../../caudate/_m/genes/diffExpr_EAvsAA_full.txt', 0.
 ↪05)
dlpfc = subset_pvalue('../../../dlpfc/_m/genes/diffExpr_EAvsAA_full.txt', 0.05)
hippo = subset_pvalue('../../../hippocampus/_m/genes/diffExpr_EAvsAA_full.txt',␣
 ↪0.05)
```

```
gyrus = subset_pvalue("../../../dentateGyrus/_m/genes/diffExpr_EAvsAA_full.txt", 0.05)

lt = list(Caudate = caudate,
          DLPFC = dlpfc,
          Hippocampus = hippo,
          `Dentate Gyrus` = gyrus)

m = make_comb_mat(lt)
cbb_palette <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
                 "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
```

```
WARNING:rpy2.rinterface_lib.callbacks:R[write to console]: Loading required
package: grid

WARNING:rpy2.rinterface_lib.callbacks:R[write to console]:
========================================
ComplexHeatmap version 2.6.2
Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
Github page: https://github.com/jokergoo/ComplexHeatmap
Documentation: http://jokergoo.github.io/ComplexHeatmap-reference

If you use it in published research, please cite:
Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
  genomic data. Bioinformatics 2016.

This message can be suppressed by:
  suppressPackageStartupMessages(library(ComplexHeatmap))
========================================
```

[19]:

```
%%R
right_annot = upset_right_annotation(
    m, ylim = c(0, 4000),
    gp = gpar(fill = "black"),
    annotation_name_side = "top",
    axis_param = list(side = "top"))

top_annot = upset_top_annotation(
    m, height=unit(7, "cm"),
    ylim = c(0, 2000),
    gp=gpar(fill=cbb_palette[comb_degree(m)]),
    annotation_name_rot = 90)

pdf('BrainSeq_race_tissue_upsetR_DEgenes.pdf', width=8, height=4)
ht = draw(UpSet(m, pt_size=unit(4, "mm"), lwd=3,
                comb_col=cbb_palette[comb_degree(m)],
                set_order = c("Caudate", "DLPFC", "Hippocampus", "Dentate␣
 ↪Gyrus"),
                comb_order = order(-comb_size(m)),
                row_names_gp = gpar(fontsize = 14, fontface='bold'),
                right_annotation = right_annot,
                top_annotation = top_annot))
od = column_order(ht)
cs = comb_size(m)
decorate_annotation("intersection_size", {
    grid.text(cs[od], x = seq_along(cs), y = unit(cs[od], "native") +
              unit(6, "pt"),
        default.units = "native", just = "bottom", gp = gpar(fontsize = 11))
})
dev.off()

svg('BrainSeq_race_tissue_upsetR_DEgenes.svg', width=8, height=4)
ht = draw(UpSet(m, pt_size=unit(4, "mm"), lwd=3,
                comb_col=cbb_palette[comb_degree(m)],
                set_order = c("Caudate", "DLPFC", "Hippocampus", "Dentate␣
 ↪Gyrus"),
                comb_order = order(-comb_size(m)),
                row_names_gp = gpar(fontsize = 14, fontface='bold'),
                right_annotation = right_annot,
                top_annotation = top_annot))
od = column_order(ht)
cs = comb_size(m)
decorate_annotation("intersection_size", {
    grid.text(cs[od], x = seq_along(cs), y = unit(cs[od], "native") +
              unit(6, "pt"),
        default.units = "native", just = "bottom", gp = gpar(fontsize = 11))
})
dev.off()
```

```
png
  2
```

[20]:

```
%%R
right_ha = rowAnnotation(
    "Intersection\nsize" = anno_barplot(comb_size(m), border=F,
                                        ylim = c(0, 2000),
                                       ␣
 ↪gp=gpar(fill=cbb_palette[comb_degree(m)]),
                                        width = unit(7, "cm")))
top_ha = HeatmapAnnotation(
    "Set size" = anno_barplot(set_size(m), border=F,
                              ylim = c(0, 4000),
                              gp = gpar(fill = "black"),
                              height = unit(2, "cm")),
    gap = unit(2, "mm"), annotation_name_side = "left",
    annotation_name_rot = 90)


pdf("BrainSeq_race_tissue_upsetR_DEgenes_transpose.pdf", width=5, height=10)
ht = draw(UpSet(t(m), pt_size=unit(5, "mm"), lwd=3,
                comb_order = order(-comb_size(m)),
                comb_col=cbb_palette[comb_degree(m)],
                set_order = c("Caudate", "DLPFC", "Hippocampus", "Dentate␣
 ↪Gyrus"),
                column_names_gp = gpar(fontsize = 16, fontface='bold'),
                right_annotation = right_ha, top_annotation=top_ha))

od = rev(row_order(ht))
cs = comb_size(m)
decorate_annotation("Intersection\nsize", {
    grid.text(cs[od], y = seq_along(cs), x = unit(cs[od], "native") +
              unit(6, "pt"),
        default.units = "native", just = "left", gp = gpar(fontsize = 11))
})
dev.off()

svg("BrainSeq_race_tissue_upsetR_DEgenes_transpose.svg", width=5, height=10)
ht = draw(UpSet(t(m), pt_size=unit(5, "mm"), lwd=3,
                comb_order = order(-comb_size(m)),
                comb_col=cbb_palette[comb_degree(m)],
                set_order = c("Caudate", "DLPFC", "Hippocampus", "Dentate␣
 ↪Gyrus"),
                column_names_gp = gpar(fontsize = 16, fontface='bold'),
                right_annotation = right_ha, top_annotation=top_ha))

od = rev(row_order(ht))
cs = comb_size(m)
decorate_annotation("Intersection\nsize", {
    grid.text(cs[od], y = seq_along(cs), x = unit(cs[od], "native") +
              unit(6, "pt"),
        default.units = "native", just = "left", gp = gpar(fontsize = 11))
})
dev.off()
```

```
png
  2
```

1.5 Annotate with gene information

[21]:
```
dft = caudate.merge(gtf_annot[['gene_id', 'gene_name', 'seqname']],
                    left_index=True, right_on='gene_id')
dft.head()
```

[21]:
```
                   Feature        ensemblID     adj.P.Val     logFC  \
2450534   ENSG00000272977.1  ENSG00000272977  1.293546e-22  2.197155
782182    ENSG00000233913.7  ENSG00000233913  1.511451e-22 -2.941671
1784411   ENSG00000259479.6  ENSG00000259479  2.536508e-22 -2.338783
295752   ENSG00000068654.15  ENSG00000068654  6.364724e-22  0.292087
47391     ENSG00000084628.9  ENSG00000084628  9.739085e-21  1.891807

                 t  Dir             gene_id      gene_name seqname
2450534  12.328222  1.0   ENSG00000272977.1  CTA-390C10.10   chr22
782182  -12.213021 -1.0   ENSG00000233913.7   CTC-575D19.1    chr5
1784411 -12.087500 -1.0   ENSG00000259479.6         SORD2P   chr15
295752   11.922914  1.0  ENSG00000068654.15         POLR1A    chr2
47391    11.518655  1.0   ENSG00000084628.9         NKAIN1    chr1
```

[22]:
```
shared_df = dft.loc[:, ['gene_id', 'ensemblID', 'seqname', 'gene_name', 'Dir']]\
        .merge(pd.DataFrame({'ensemblID': list(set(phase2_dlpfc['ensemblID']) &
                                               set(phase2_hippo['ensemblID']) &
                                               set(phase3_caudate['ensemblID'])␣
 ↪&
                                                                                 ␣
 ↪set(dentate_gyrus['ensemblID']))}),
               on='ensemblID')
shared_df.to_csv('BrainSeq_shared_degs_annotation.txt',
                 sep='\t', index=False, header=True)
shared_df.head()
```

[22]:
```
              gene_id        ensemblID seqname      gene_name  Dir
0   ENSG00000272977.1  ENSG00000272977   chr22  CTA-390C10.10  1.0
1   ENSG00000233913.7  ENSG00000233913    chr5   CTC-575D19.1 -1.0
2   ENSG00000259479.6  ENSG00000259479   chr15         SORD2P -1.0
3  ENSG00000068654.15  ENSG00000068654    chr2         POLR1A  1.0
4   ENSG00000226278.1  ENSG00000226278    chr7         PSPHP1 -1.0
```

[23]:
```
dd = np.sum(shared_df.seqname.isin(['chrX', 'chrY'])) / shared_df.shape[0] * 100
print("%0.2f%% of shared DEG are allosomal!" % dd)
```

```
4.55% of shared DEG are allosomal!
```

[24]:
```
gtf_annot['ensemblID'] = gtf_annot.gene_id.str.replace("\\..*", "")
gtf_annot[['gene_id', 'ensemblID', 'gene_name', 'seqname', 'gene_type']]\
    .merge(df, left_on='ensemblID', right_index=True)\
    .to_csv('brainseq_deg_across_tissues_comparison.csv', index=False)
```

```
<ipython-input-1-4f417e935742>:1: FutureWarning: The default value of regex will
change from True to False in a future version.
  gtf_annot['ensemblID'] = gtf_annot.gene_id.str.replace("\\..*", "")
```

[ ]:

```

```