main_female

November 24, 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
```

[2]:

```python
config = {
    'caudate': '../../../caudate/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_4features.txt',
    'dlpfc': '../../../dlpfc/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_4features.txt',
    'hippo': '../../../hippocampus/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_4features.txt',
    'cmc_dlpfc': '../../../cmc_dlpfc/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_genes.txt'
}
```

[3]:

```python
@functools.lru_cache()
def get_deg(filename):
    dft = pd.read_csv(filename, sep='\t', index_col=0)
    dft = dft[(dft['Type'] == 'gene')].copy()
    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)
    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])
```

[4]:
```
def tissue_annotation(tissue):
    return {'dlpfc': 'DLPFC', 'hippo': 'Hippocampus',
            'caudate': 'Caudate', 'cmc_dlpfc': 'CMC DLPFC'}[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)
```

1.1 BrainSeq Comparison

[5]:
```
caudate = get_deg(config['caudate'])
caudate.groupby('Dir').size()
```

[5]:
```
Dir
-1.0    16
 1.0    14
dtype: int64
```

[6]:
```
caudate[(caudate['adj.P.Val'] < 0.05)].shape
```

```
INFO:numexpr.utils:Note: NumExpr detected 60 cores but "NUMEXPR_MAX_THREADS" not
set, so enforcing safe limit of 8.
INFO:numexpr.utils:NumExpr defaulting to 8 threads.
```

[6]:
```
(30, 6)
```

[7]:
```
dlpfc = get_deg(config['dlpfc'])
dlpfc.groupby('Dir').size()
```

[7]:
```
Series([], dtype: int64)
```

[8]:
```
dlpfc[(dlpfc['adj.P.Val'] < 0.05)].shape
```

[8]:
```
(0, 6)
```

[9]:

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

[9]:

```
Series([], dtype: int64)
```

[10]:

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

[10]:

```
(0, 6)
```

1.1.1 Upset Plot

[11]:

```
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']]
```

[12]:

```
geneList = pd.merge(phase3_caudate[['ensemblID']], phase2_dlpfc[['ensemblID']],
                    on=['ensemblID'], how='outer')\
             .merge(phase2_hippo[['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')

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

```
(30, 2) (30, 2) (30, 2)
```

[13]:

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

[13]:

```
                 Caudate  DLPFC  Hippocampus
ensemblID
ENSG00000003137        1      0            0
ENSG00000070915        1      0            0
```

[14]:

```
%load_ext rpy2.ipython
```

[15]:

```
%%R
library(ComplexHeatmap)
library(tidyverse)
subset_pvalue <- function(filename, fdr_cutoff){
    df <- data.table::fread(filename) %>%
        filter(Type == 'gene', adj.P.Val < fdr_cutoff)
    return(df$ensemblID)
}

caudate = subset_pvalue('../../../caudate/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_4features.txt',
                        0.05)
dlpfc = subset_pvalue('../../../dlpfc/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_4features.txt',
                      0.05)
hippo = subset_pvalue('../../../hippocampus/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_4features.txt',
                      0.05)

lt = list(Caudate = caudate,
          DLPFC = dlpfc,
          Hippocampus = hippo)

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 2.10.0
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.
```

```
The new InteractiveComplexHeatmap package can directly export static
complex heatmaps into an interactive Shiny app with zero effort. Have a try!

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


WARNING:rpy2.rinterface_lib.callbacks:R[write to console]:   Attaching packages
                      tidyverse 1.3.1

WARNING:rpy2.rinterface_lib.callbacks:R[write to console]:  ggplot2 3.3.5
purrr   0.3.4
 tibble  3.1.6      dplyr   1.0.7
 tidyr   1.1.4      stringr 1.4.0
 readr   2.1.0      forcats 0.5.1

WARNING:rpy2.rinterface_lib.callbacks:R[write to console]:   Conflicts
                       tidyverse_conflicts()
 dplyr::filter() masks stats::filter()
 dplyr::lag()    masks stats::lag()
```

[16]:

```
%%R
right_annot = upset_right_annotation(
    m, ylim = c(0, 150),
    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, 150),
    gp=gpar(fill=cbb_palette[comb_degree(m)]),
    annotation_name_rot = 90)

pdf('BrainSeq_sex_tissue_upsetR_DEgenes_femaleSpecific.pdf', width=6, 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"),
                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
```

[17]:
```
%%R
right_ha = rowAnnotation(
    "Intersection\nsize" = anno_barplot(comb_size(m), border=F,
                                        ylim = c(0, 150),
                                       ␣
 ↪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, 150),
                              gp = gpar(fill = "black"),
                              height = unit(2, "cm")),
    gap = unit(2, "mm"), annotation_name_side = "left",
    annotation_name_rot = 90)


pdf("BrainSeq_sex_tissue_upsetR_DEgenes_transpose_femaleSpecific.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"),
                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
```

1.1.2 Shared features

[18]:

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

[19]:

```
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"]]

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']
```

[19]:

```
              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      -
```

[20]:

```
dft = caudate.merge(gtf_annot[['gene_id', 'gene_name', 'seqname']],
                    left_index=True, right_on='gene_id')
dft.head(2)
```

[20]:

```
                    Feature        ensemblID  adj.P.Val     logFC         t  \
1928121   ENSG00000070915.9  ENSG00000070915   0.006380  0.909953  4.668428
1468999  ENSG00000111181.12  ENSG00000111181   0.009994 -0.432628 -4.405278

         Dir             gene_id gene_name seqname
1928121  1.0   ENSG00000070915.9   SLC12A3   chr16
1468999 -1.0  ENSG00000111181.12   SLC6A12   chr12
```

1.2 Comparison with CommonMind

[21]:

```
cmc_dlpfc = get_deg(config['cmc_dlpfc'])
cmc_dlpfc.groupby('Dir').size()
```

[21]:

```
Dir
-1.0    227
```

```
 1.0    356
dtype: int64
```

[22]:
```
cmc_dlpfc[(cmc_dlpfc['adj.P.Val'] < 0.05)].shape
```

[22]:
```
(583, 6)
```

1.2.1 Upset Plot

[23]:
```
cmc = cmc_dlpfc[(cmc_dlpfc['adj.P.Val'] < 0.05)].copy()
cmc['CMC DLPFC'] = 1
cmc = cmc[['ensemblID', 'CMC DLPFC']].groupby('ensemblID').first().reset_index()
```

[24]:
```
geneList = pd.merge(phase3_caudate[['ensemblID']], phase2_dlpfc[['ensemblID']],␣
 ↪on=['ensemblID'], how='outer')\
             .merge(phase2_hippo[['ensemblID']], on=['ensemblID'], how='outer')\
             .merge(cmc[['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')

newCMC = pd.merge(geneList, cmc, on=['ensemblID'], how='outer').fillna(0)
newCMC['CMC DLPFC'] = newCMC['CMC DLPFC'].astype('int')

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

```
(610, 2) (610, 2) (610, 2) (610, 2)
```

[25]:
```
df = pd.concat([newC.set_index(['ensemblID']), newD1.set_index(['ensemblID']),
                newH.set_index(['ensemblID']), newCMC.set_index(['ensemblID'])],
               axis=1, join='outer')
df.head(2)
```

[25]:
```
                 Caudate  DLPFC  Hippocampus  CMC DLPFC
ensemblID
ENSG00000003137        1      0            0          0
ENSG00000003147        0      0            0          1
```

[26]:
```
%%R
cmc = subset_pvalue('../../../cmc_dlpfc/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_genes.txt',
                    0.05)

lt = list(Caudate = caudate,
          DLPFC = dlpfc,
          Hippocampus = hippo,
          `CMC DLPFC` = cmc)

m = make_comb_mat(lt)
```

[27]:
```
%%R
right_annot = upset_right_annotation(
    m, ylim = c(0, 800),
    gp = gpar(fill = "black"),
    annotation_name_side = "bottom",
    axis_param = list(side = "bottom"))

top_annot = upset_top_annotation(
    m, height=unit(7, "cm"),
    ylim = c(0, 800),
    gp=gpar(fill=cbb_palette[comb_degree(m)]),
    annotation_name_rot = 90)

pdf('cmc_sex_tissue_upsetR_DEgenes_femaleSpecific.pdf', width=8, height=5)
ht = draw(UpSet(m, pt_size=unit(6, "mm"), lwd=3,
                comb_col=cbb_palette[comb_degree(m)],
                set_order = c("Caudate", "DLPFC", "Hippocampus", "CMC DLPFC"),
                comb_order = order(-comb_size(m)),
                row_names_gp = gpar(fontsize = 16, 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
```

[28]:
```
%%R
right_ha = rowAnnotation(
    "Intersection\nsize" = anno_barplot(comb_size(m), border=F,
                                        ylim = c(0, 800),
                                       ␣
 ↪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, 800),
                              gp = gpar(fill = "black"),
                              height = unit(2, "cm")),
    gap = unit(2, "mm"), annotation_name_side = "left",
    annotation_name_rot = 90)

pdf("cmc_sex_tissue_upsetR_DEgenes_transpose_femaleSpecific.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", "CMC DLPFC"),
                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
```

[29]:
```
dft = pd.read_csv('../../../cmc_dlpfc/female_analysis/metrics_summary/_m/
 ↪female_specific_DE_genes.txt',
                  sep='\t')
dft['Dir'] = np.sign(dft['t'])
dft.head()
```

[29]:
```
              Feature            gencodeID     Symbol         ensemblID  Chrom  \
0  ENSG00000153132.12  ENSG00000153132.12       CLGN   ENSG00000153132   chr4
1   ENSG00000179083.6   ENSG00000179083.6    FAM133A   ENSG00000179083   chrX
2   ENSG00000165733.7   ENSG00000165733.7       BMS1   ENSG00000165733  chr10
3  ENSG00000183023.18  ENSG00000183023.18     SLC8A1   ENSG00000183023   chr2
4   ENSG00000236268.5   ENSG00000236268.5  LINC01361   ENSG00000236268   chr1
```

```
      logFC         t  adj.P.Val  Male_Pval  Male_FDR  Type  Dir
0  0.389937  5.559139   0.000123   0.283020  0.313711  gene  1.0
1  0.261268  5.004488   0.000535   0.272019  0.302973  gene  1.0
2  0.150918  4.986552   0.000535   0.169787  0.205428  gene  1.0
3  0.245819  4.925477   0.000632   0.082759  0.108963  gene  1.0
4  0.404532  4.865744   0.000700   0.110632  0.140404  gene  1.0
```

[30]:

```
dft.loc[:, ['Feature', 'ensemblID', 'Symbol', 'Chrom', 'Dir']]\
    .merge(pd.DataFrame({'ensemblID': list(set(phase3_caudate['ensemblID']) &
                                           set(cmc['ensemblID']))}),
           on='ensemblID')
```

[30]:

```
              Feature        ensemblID   Symbol  Chrom  Dir
0   ENSG00000263006.6  ENSG00000263006  ROCK1P1  chr18  1.0
1   ENSG00000249669.9  ENSG00000249669    CARMN   chr5 -1.0
2  ENSG00000167703.14  ENSG00000167703  SLC43A2  chr17 -1.0
```

[31]:

```
shared_df = dft.loc[:, ['Feature', 'ensemblID', 'Chrom', 'Symbol', 'Dir']]\
               .merge(pd.DataFrame({'ensemblID':␣
 ↪list(set(phase3_caudate['ensemblID']) &
                                                      set(cmc['ensemblID']))}),
                      on='ensemblID')
shared_df.to_csv('cmc_shared_caudate_degs_annotation_femaleSpecific.txt',␣
 ↪sep='\t',
                 index=False, header=True)
shared_df
```

[31]:

```
              Feature        ensemblID  Chrom   Symbol  Dir
0   ENSG00000263006.6  ENSG00000263006  chr18  ROCK1P1  1.0
1   ENSG00000249669.9  ENSG00000249669   chr5    CARMN -1.0
2  ENSG00000167703.14  ENSG00000167703  chr17  SLC43A2 -1.0
```

[32]:

```
#### 6 out of 41 are autosomal
dd = np.sum(shared_df.Chrom.isin(['chrX', 'chrY'])) / shared_df.shape[0] * 100
print("%0.2f%% of shared DEG are allosomal!" % dd)
```

```
0.00% of shared DEG are allosomal!
```

[33]:

```
gtf_annot['ensemblID'] = gtf_annot.gene_id.str.replace("\\..*", "", regex=True)
gtf_annot[["gene_id", 'ensemblID', 'gene_name', 'seqname', 'gene_type']]\
    .merge(df, left_on='ensemblID', right_index=True)\
    .to_csv('cmc_all_deg_across_tissues_femaleSpecific.csv')
```

[ ]: