main

July 11, 2021

1 Tissue comparison for differential expression analysis

[1]:

```
import functools
import numpy as np
import pandas as pd
```

[2]:

```
config = {
    'caudate': '../../../caudate/_m/genes/diffExpr_maleVfemale_full.txt',
    'dlpfc': '../../../dlpfc/_m/genes/diffExpr_maleVfemale_full.txt',
    'hippo': '../../../hippocampus/_m/genes/diffExpr_maleVfemale_full.txt',
    'cmc_dlpfc': '../../../cmc_dlpfc/_m/genes/diffExpr_maleVfemale_full.txt',
}
```

[3]:

```
@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])
```

[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    11133
 1.0    12355
dtype: int64
```

[6]:

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

[6]:

```
(380, 6)
```

[7]:

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

[7]:

```
Dir
-1.0    11240
 1.0    11799
dtype: int64
```

[8]:

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

[8]:

```
(573, 6)
```

[9]:

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

[9]:

```
Dir
-1.0    11840
 1.0    11150
dtype: int64
```

[10]:

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

[10]:

```
(105, 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)
```

```
(848, 2) (848, 2) (848, 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
ENSG00000002586         1      1            1
ENSG00000003137         1      0            0
```

[14]:
```
df.to_csv('brainseq_deg_across_tissues_comparison.csv')
```

[15]:
```
%load_ext rpy2.ipython
```

[16]:
```
%%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_maleVfemale_full.
 ↪txt', 0.05)
dlpfc = subset_pvalue('../../../dlpfc/_m/genes/diffExpr_maleVfemale_full.txt',␣
 ↪0.05)
hippo = subset_pvalue('../../../hippocampus/_m/genes/diffExpr_maleVfemale_full.
 ↪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")
```

```
R[write to console]: Loading required package: grid

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))
========================================
```

[17]:

```
%%R
right_annot = upset_right_annotation(
    m, ylim = c(0, 550),
    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, 500),
    gp=gpar(fill=cbb_palette[comb_degree(m)]),
    annotation_name_rot = 90)

pdf('BrainSeq_sex_tissue_upsetR_DEgenes.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()

svg('BrainSeq_sex_tissue_upsetR_DEgenes.svg', 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
```

[18]:
```
%%R
right_ha = rowAnnotation(
    "Intersection\nsize" = anno_barplot(comb_size(m), border=F,
                                        ylim = c(0, 500),
                                       ␣
 ↪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, 550),
                              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.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()

svg("BrainSeq_sex_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"),
                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
```

1.2 Annotated shared genes

[19]:

```
from gtfparse import read_gtf
```

[20]:

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

[21]:

```
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']
```

[21]:

```
              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      -
```

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

```
[22]:                   Feature        ensemblID      adj.P.Val     logFC  \
      2529938  ENSG00000229807.10  ENSG00000229807  1.953623e-272 -9.296137
      2573932  ENSG00000114374.12  ENSG00000114374  1.953623e-272  8.683679
      2574263  ENSG00000183878.15  ENSG00000183878  8.133127e-253  8.597152
      2575964  ENSG00000012817.15  ENSG00000012817  3.593495e-252  8.693010
      2574099  ENSG00000067048.16  ENSG00000067048  5.035188e-250  8.587803

                        t  Dir             gene_id gene_name seqname
      2529938 -100.356075 -1.0  ENSG00000229807.10      XIST    chrX
      2573932  100.180866  1.0  ENSG00000114374.12     USP9Y    chrY
      2574263   88.466208  1.0  ENSG00000183878.15       UTY    chrY
      2575964   88.036823  1.0  ENSG00000012817.15     KDM5D    chrY
      2574099   86.796856  1.0  ENSG00000067048.16     DDX3Y    chrY
```

```
[23]: 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']))}),
                     on='ensemblID')
      shared_df.to_csv('BrainSeq_shared_degs_annotation.txt',
                       sep='\t', index=False, header=True)
      shared_df.head()
```

```
[23]:               gene_id        ensemblID seqname gene_name  Dir
      0  ENSG00000229807.10  ENSG00000229807    chrX      XIST -1.0
      1  ENSG00000114374.12  ENSG00000114374    chrY     USP9Y  1.0
      2  ENSG00000183878.15  ENSG00000183878    chrY       UTY  1.0
      3  ENSG00000012817.15  ENSG00000012817    chrY     KDM5D  1.0
      4  ENSG00000067048.16  ENSG00000067048    chrY     DDX3Y  1.0
```

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

```
69.86% of shared DEG are allosomal!
```

1.3 Comparison with CommonMind

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

[25]:
```
Dir
-1.0    10915
 1.0     9705
dtype: int64
```

[26]:
```
cmc_dlpfc[(cmc_dlpfc['adj.P.Val'] < 0.05)].shape
```

[26]:
```
(1315, 6)
```

1.3.1 Upset Plot

[27]:
```
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()
```

[28]:
```
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)
```

```
(2066, 2) (2066, 2) (2066, 2) (2066, 2)
```

[29]:
```
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)
```

[29]:
```
                 Caudate  DLPFC  Hippocampus  CMC DLPFC
ensemblID
ENSG00000000460        0      0            0          1
ENSG00000001460        0      0            0          1
```

[30]:

```
df.to_csv('cmc_all_deg_across_tissues.csv')
```

[31]:

```
%%R
library(tidyverse)
subset_pvalue <- function(fn, fdr_cutoff){
    df <- data.table::fread(fn) %>% filter(adj.P.Val < 0.05)
    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)
}
```

```
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  3.1.2      dplyr   1.0.7
 tidyr   1.1.3      stringr 1.4.0
 readr   1.4.0      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()
```

[32]:

```
%%R
#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)
cmc = subset_pvalue('../../../cmc_dlpfc/_m/genes/diffExpr_maleVfemale_full.
 ↪txt', 0.05)

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

m = make_comb_mat(lt)
```

[33]:

```
%%R
right_annot = upset_right_annotation(
    m, ylim = c(0, 1500),
    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, 1500),
    gp=gpar(fill=cbb_palette[comb_degree(m)]),
    annotation_name_rot = 90)

pdf('cmc_sex_tissue_upsetR_DEgenes.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()

svg('cmc_sex_tissue_upsetR_DEgenes.svg', 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
```

[34]:

```
%%R
right_ha = rowAnnotation(
    "Intersection\nsize" = anno_barplot(comb_size(m), border=F,
                                        ylim = c(0, 1500),
                                       ␣
 ↪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, 1500),
                              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.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()

svg("cmc_sex_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", "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
```

```
[35]: dft = pd.read_csv('../../../cmc_dlpfc/metrics_summary/_m/chromosome_DEG.csv',␣
       ↪index_col=0)
      dft['Feature'] = dft.index
      dft['Dir'] = np.sign(dft['t'])
      dft['ensemblID'] = dft.index.str.replace('\\..*', '', regex=True)
      dft.head()
```

```
[35]:                           logFC   AveExpr         t       P.Value  adj.P.Val  \
      gene_id
      ENSG00000174796.12 -0.142817  3.601461 -5.005702  7.696597e-07    0.01523
      ENSG00000198873.11  0.174953  3.674668  4.765750  2.462456e-06    0.01523
      ENSG00000164620.8   0.251649  4.480607  4.708131  3.232510e-06    0.01523
      ENSG00000152558.14 -0.227384  5.923397 -4.587058  5.747358e-06    0.01523
      ENSG00000128915.11 -0.153500  5.268981 -4.578421  5.903731e-06    0.01523

                             z.std  ensembl_gene_id position Direction hgnc_symbol  \
      gene_id
      ENSG00000174796.12 -4.942908  ENSG00000174796       12      DOWN       THAP6
      ENSG00000198873.11  4.711214  ENSG00000198873       11        UP        GRK5
      ENSG00000164620.8   4.655465  ENSG00000164620        8        UP       RELL2
      ENSG00000152558.14 -4.535477  ENSG00000152558       14      DOWN     TMEM123
      ENSG00000128915.11 -4.529808  ENSG00000128915       11      DOWN        ICE2

                          percentage_gene_gc_content    gene_biotype  \
      gene_id
      ENSG00000174796.12                       40.52  protein_coding
      ENSG00000198873.11                       50.47  protein_coding
      ENSG00000164620.8                        60.00  protein_coding
      ENSG00000152558.14                       40.62  protein_coding
      ENSG00000128915.11                       36.91  protein_coding

                         chromosome_name             Feature  Dir        ensemblID
      gene_id
      ENSG00000174796.12               4  ENSG00000174796.12 -1.0  ENSG00000174796
      ENSG00000198873.11              10  ENSG00000198873.11  1.0  ENSG00000198873
      ENSG00000164620.8                5   ENSG00000164620.8  1.0  ENSG00000164620
      ENSG00000152558.14              11  ENSG00000152558.14 -1.0  ENSG00000152558
      ENSG00000128915.11              15  ENSG00000128915.11 -1.0  ENSG00000128915
```

```
[36]: shared_df = dft.rename(columns={'chromosome_name': 'seqname',
                                      'hgnc_symbol': 'gene_name'})\
                     .loc[:, ['Feature', 'ensemblID', 'seqname', 'gene_name', 'Dir']]\
                     .merge(pd.DataFrame({'ensemblID':␣
       ↪list(set(phase2_dlpfc['ensemblID']) &
                                                           ␣
      ↪set(phase2_hippo['ensemblID']) &
                                                           ␣
      ↪set(phase3_caudate['ensemblID']) &
                                                            set(cmc['ensemblID']))}),
                            on='ensemblID')
      shared_df.seqname = 'chr'+shared_df.seqname
      shared_df.to_csv('cmc_shared_degs_annotation.txt', sep='\t', index=False,␣
      ↪header=True)
      shared_df
```

[36]:

```
              Feature        ensemblID seqname gene_name  Dir
0  ENSG00000147050.14  ENSG00000147050    chrX     KDM6A -1.0
```

[37]:

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

```
100.00% of shared DEG are allosomal!
```

[ ]:

```

```