main

September 18, 2021

1 Tissue comparison for differential expression analysis

[1]:
```
import functools
import numpy as np
import pandas as pd
from plotnine import *
from gtfparse import read_gtf
from warnings import filterwarnings
from matplotlib.cbook import mplDeprecation
filterwarnings('ignore', category=mplDeprecation)
filterwarnings('ignore', category=UserWarning, module='plotnine.*')
filterwarnings('ignore', category=DeprecationWarning, module='plotnine.*')
```

[2]:
```
config = {
    'caudate': '../../../_m/genes/diffExpr_szVctl_full.txt',
    'dlpfc': '/ceph/projects/v4_phase3_paper/inputs/public_data/_m/phase2/
 ↪dlpfc_diffExpr_szVctl_full.txt',
    'hippo': '/ceph/projects/v4_phase3_paper/inputs/public_data/_m/phase2/
 ↪hippo_diffExpr_szVctl_full.txt',
    'cmc_sva': '/ceph/projects/v4_phase3_paper/inputs/public_data/_m/cmc/
 ↪CMC_MSSM-Penn-Pitt_DLPFC_mRNA_'+\
   ␣
 ↪'IlluminaHiSeq2500_gene-adjustedSVA-differentialExpression-includeAncestry-DxSCZ-DE.
 ↪tsv',
    'cmc': '/ceph/projects/v4_phase3_paper/inputs/public_data/_m/cmc/
 ↪CMC_MSSM-Penn-Pitt_DLPFC_mRNA_'+\
   ␣
 ↪'IlluminaHiSeq2500_gene-adjustedNoSVA-differentialExpression-includeAncestry-DxSCZ-DE.
 ↪tsv'
}
```

[3]:
```
@functools.lru_cache()
def get_gtf(gtf_file):
    return read_gtf(gtf_file)


@functools.lru_cache()
```

```
def get_cmc(SVA=True):
    if SVA:
        cmc_dlpfc = pd.read_csv(config["cmc_sva"], sep='\t')\
                      .rename(columns={'MAPPED_genes': 'Symbol', "genes":␣
 ↪"ensemblID"})
    else:
        cmc_dlpfc = pd.read_csv(config["cmc"], sep='\t')\
                      .rename(columns={'MAPPED_genes': "Symbol", "genes":␣
 ↪"ensemblID"})
    cmc_dlpfc['Dir'] = np.sign(cmc_dlpfc['t'])
    cmc_dlpfc["Feature"] = cmc_dlpfc.ensemblID
    return cmc_dlpfc[["Feature", "ensemblID", 'adj.P.Val', 't', 'Dir',␣
 ↪"Symbol"]]


@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': "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)


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"]]
```

1.1 BrainSeq Comparison

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

[5]:
```
Dir
-1.0    12061
 1.0    10897
dtype: int64
```

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

```
INFO:numexpr.utils:Note: NumExpr detected 64 cores but "NUMEXPR_MAX_THREADS" not
set, so enforcing safe limit of 8.
INFO:numexpr.utils:NumExpr defaulting to 8 threads.
```

[6]:
```
(2701, 6)
```

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

[7]:
```
Dir
-1.0    13207
 1.0    11445
dtype: int64
```

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

[8]:
```
(245, 6)
```

[9]:
```
hippo = get_deg(config['hippo'])
hippo.groupby('Dir').size()
```

[9]:
```
Dir
-1.0    12852
 1.0    11800
dtype: int64
```

[10]:
```
hippo[(hippo['adj.P.Val'] < 0.05)].shape
```

[10]:
```
(48, 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()
## Caudate
newC = pd.merge(geneList, phase3_caudate, on=['ensemblID'],
                how='outer').fillna(0)
newC['Caudate'] = newC['Caudate'].astype('int')
## DLPFC
newD1 = pd.merge(geneList, phase2_dlpfc, on=['ensemblID'],
                 how='outer').fillna(0)
newD1['DLPFC'] = newD1['DLPFC'].astype('int')
## Hippocampus
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)
```

```
(2929, 2) (2929, 2) (2929, 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]:

ensemblID	Caudate	DLPFC	Hippocampus
ENSG00000001084	1	0	0
ENSG00000001497	1	0	0

[14]:
```
%load_ext rpy2.ipython
```

[15]:
```
%%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('../../../_m/genes/diffExpr_szVctl_full.txt', 0.05)
dlpfc = subset_pvalue('/ceph/projects/v4_phase3_paper/inputs/public_data/_m/
 ↪phase2/dlpfc_diffExpr_szVctl_full.txt', 0.05)
hippo = subset_pvalue('/ceph/projects/v4_phase3_paper/inputs/public_data/_m/
 ↪phase2/hippo_diffExpr_szVctl_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")
```

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

[16]:
```
%%R
right_annot = upset_right_annotation(
    m, ylim = c(0, 3000),
    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, 3000),
    gp=gpar(fill=cbb_palette[comb_degree(m)]),
    annotation_name_rot = 90)

pdf('BrainSeq_dx_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"),
                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_dx_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"),
                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, 3000),
 ↪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, 3000),
                              gp = gpar(fill = "black"),
                              height = unit(2, "cm")),
    gap = unit(2, "mm"), annotation_name_side = "left",
    annotation_name_rot = 90)


pdf("BrainSeq_dx_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_dx_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
```

[18]:
```
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']
```

[18]:
```
              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      -
```

[19]:
```
dft = caudate.merge(gtf_annot[['gene_id', 'gene_name', 'seqname']],
                    left_index=True, right_on='gene_id')
dft.head()
```

[19]:
```
                    Feature        ensemblID     adj.P.Val     logFC  \
699321    ENSG00000248587.7  ENSG00000248587  1.387742e-26  0.801502
2481489   ENSG00000138944.7  ENSG00000138944  1.707516e-24  0.563733
```

```
2362233  ENSG00000185052.11  ENSG00000185052  3.972599e-21  0.291763
1705114  ENSG00000140015.19  ENSG00000140015  6.716497e-18  0.515655
2551916  ENSG00000171004.17  ENSG00000171004  3.196203e-16  0.302105

                t  Dir            gene_id gene_name seqname
699321   12.696887  1.0   ENSG00000248587.7  GDNF-AS1    chr5
2481489  12.073351  1.0   ENSG00000138944.7  KIAA1644   chr22
2362233  11.122852  1.0  ENSG00000185052.11   SLC24A3   chr20
1705114  10.185331  1.0  ENSG00000140015.19     KCNH5   chr14
2551916   9.670025  1.0  ENSG00000171004.17    HS6ST2    chrX
```

[20]:
```
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()
```

[20]:
```
Empty DataFrame
Columns: [gene_id, ensemblID, seqname, gene_name, Dir]
Index: []
```

[21]:
```
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('brainseq_deg_across_tissues_comparison.csv', index=False)
```

1.2 CMC Comparison

1.2.1 Adjusted SVA

[22]:
```
cmc = get_cmc(SVA=True)
cmc.groupby('Dir').size()
```

[22]:
```
Dir
-1.0    8898
 1.0    7525
dtype: int64
```

[23]:
```
cmc[(cmc['adj.P.Val'] < 0.05)].shape
```

[23]:
```
(419, 6)
```

1.2.2 No adjusted SVA

[24]:
```
cmc_dlpfc2 = get_cmc(False)
cmc_dlpfc2.groupby('Dir').size()
```

[24]:
```
Dir
-1.0    8759
 1.0    7664
dtype: int64
```

[25]:
```
cmc_dlpfc2[(cmc_dlpfc2['adj.P.Val'] < 0.05)].shape
```

[25]:
```
(573, 6)
```

1.2.3 Upset Plot

[26]:
```
cmc_dlpfc = cmc[(cmc['adj.P.Val'] < 0.05)].copy()
cmc_dlpfc['CMC DLPFC'] = 1
cmc_dlpfc = cmc_dlpfc[['ensemblID', 'CMC DLPFC']]

cmc_noSva = cmc_dlpfc2[(cmc_dlpfc2['adj.P.Val'] < 0.05)].copy()
cmc_noSva['CMC DLPFC (no SVA)'] = 1
cmc_noSva = cmc_noSva[['ensemblID', 'CMC DLPFC (no SVA)']]

geneList = pd.merge(phase3_caudate[['ensemblID']],
                    cmc_dlpfc[['ensemblID']],
                    on=['ensemblID'], how='outer')\
              .merge(cmc_noSva[['ensemblID']],
                    on=['ensemblID'], how='outer')\
              .groupby(['ensemblID']).first().reset_index()
## Caudate
newC = pd.merge(geneList, phase3_caudate, on=['ensemblID'],
                how='outer').fillna(0)
newC['Caudate'] = newC['Caudate'].astype('int')
## CMC DLPFC
newD2 = pd.merge(geneList, cmc_dlpfc, on=['ensemblID'],
                 how='outer').fillna(0)
newD2['CMC DLPFC'] = newD2['CMC DLPFC'].astype('int')
## Hippocampus
newD3 = pd.merge(geneList, cmc_noSva, on=['ensemblID'],
                how='outer').fillna(0)
newD3['CMC DLPFC (no SVA)'] = newD3['CMC DLPFC (no SVA)'].astype('int')

print(newC.shape, newD2.shape, newD3.shape)
```

```
(3418, 2) (3418, 2) (3418, 2)
```

[27]:
```
df = pd.concat([newC.set_index(['ensemblID']),
                newD2.set_index(['ensemblID']),
                newD3.set_index(['ensemblID'])],
               axis=1, join='outer')
df.head(2)
```

[27]:

ensemblID	Caudate	CMC DLPFC	CMC DLPFC (no SVA)
ENSG00000001084	1	0	0
ENSG00000001497	1	0	0

[28]:
```
%%R
library(dplyr)
subset_cmc <- function(filename, fdr_cutoff){
    df <- data.table::fread(filename) %>% filter(`adj.P.Val` < 0.05) %>%
        mutate(ensemblID=genes)
    return(df$ensemblID)
}

caudate = subset_pvalue('../../../_m/genes/diffExpr_szVctl_full.txt', 0.05)
cmc_dlpfc = subset_cmc(paste0("/ceph/projects/v4_phase3_paper/inputs/
 ↪public_data/_m/cmc/CMC_MSSM-Penn-Pitt_DLPFC_mRNA_",
                               ␣
 ↪"IlluminaHiSeq2500_gene-adjustedSVA-differentialExpression-includeAncestry-DxSCZ-DE.
 ↪tsv"),
                        0.05)
cmc_noSVA = subset_cmc(paste0("/ceph/projects/v4_phase3_paper/inputs/
 ↪public_data/_m/cmc/CMC_MSSM-Penn-Pitt_DLPFC_mRNA_",
                               ␣
 ↪"IlluminaHiSeq2500_gene-adjustedNoSVA-differentialExpression-includeAncestry-DxSCZ-DE.
 ↪tsv"),
                        0.05)

lt = list(Caudate = caudate,
          `CMC DLPFC` = cmc_dlpfc,
          `CMC DLPFC (no SVA)` = cmc_noSVA)

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]:
Attaching package: ‘dplyr’


WARNING:rpy2.rinterface_lib.callbacks:R[write to console]: The following objects
are masked from ‘package:stats’:
```

```
    filter, lag


WARNING:rpy2.rinterface_lib.callbacks:R[write to console]: The following objects
are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

[29]:
```
%%R
right_annot = upset_right_annotation(
    m, ylim = c(0, 3000),
    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, 3000),
    gp=gpar(fill=cbb_palette[comb_degree(m)]),
    annotation_name_rot = 90)

pdf('CMC_dx_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", "CMC DLPFC", "CMC DLPFC (no SVA)"),
                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('CMC_dx_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", "CMC DLPFC", "CMC DLPFC (no SVA)"),
                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
```

[30]:
```
%%R
right_ha = rowAnnotation(
    "Intersection\nsize" = anno_barplot(comb_size(m), border=F,
                                        ylim = c(0, 3000),

 ↪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, 3000),
                              gp = gpar(fill = "black"),
                              height = unit(2, "cm")),
    gap = unit(2, "mm"), annotation_name_side = "left",
    annotation_name_rot = 90)


pdf("CMC_dx_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", "CMC DLPFC", "CMC DLPFC (no SVA)"),
                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_dx_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", "CMC DLPFC", "CMC DLPFC (no SVA)"),
                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
```

[ ]: