

# main\_male

November 24, 2021

## 1 Tissue comparison for differential expression analysis

```
[1]: import functools
import numpy as np
import pandas as pd
from gtfparse import read_gtf

[2]: config = {
    'caudate': '../..../caudate/male_analysis/metrics_summary/_m/
↪male_specific_DE_4features.txt',
    'dlpfc': '../..../dlpfc/male_analysis/metrics_summary/_m/
↪male_specific_DE_4features.txt',
    'hippo': '../..../hippocampus/male_analysis/metrics_summary/_m/
↪male_specific_DE_4features.txt',
    'cmc_dlpfc': '../..../cmc_dlpfc/male_analysis/metrics_summary/_m/
↪male_specific_DE_genes.txt'
}

[3]: @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 = 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)

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    718
     1.0   1140
    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]: (1858, 6)

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

```
[7]: Dir  
     -1.0    66  
      1.0    56  
      dtype: int64
```

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

[8]: (122, 6)

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

```
[9]: Dir  
     -1.0    62  
      1.0    42  
      dtype: int64
```

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

[10]: (104, 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_caodate, on=['ensemblID'], how='outer').
↳fillna(0)
newC['Caodate'] = newC['Caodate'].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)

```

(2051, 2) (2051, 2) (2051, 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]:
           Caodate  DLPFC  Hippocampus
ensemblID
ENSG00000000971      1      0           0
ENSG00000002330      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)
}

caodate = subset_pvalue('../.../caodate/male_analysis/metrics_summary/_m/
↳male_specific_DE_4features.txt',
                        0.05)
dlpfc = subset_pvalue('../.../dlpfc/male_analysis/metrics_summary/_m/
↳male_specific_DE_4features.txt',
                      0.05)
hippo = subset_pvalue('../.../hippocampus/male_analysis/metrics_summary/_m/
↳male_specific_DE_4features.txt',
                      0.05)

lt = list(Caodate = caodate,
          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, 2000),
  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_sex_tissue_upsetR_DEgenes_maleSpecific.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()

```

png  
2

```

[17]: %%%R
right_ha = rowAnnotation(
  "Intersection\ncsize" = 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, 2000),
    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_maleSpecific.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]: 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(2)

```

```

[19]:
      Feature      ensemblID  adj.P.Val  logFC \
424344  ENSG00000188011.5  ENSG00000188011  2.218141e-08 -0.387309
1080714  ENSG00000205268.10  ENSG00000205268  2.640453e-07  0.153181

      t  Dir      gene_id gene_name seqname
424344 -6.866649 -1.0  ENSG00000188011.5   RTP5   chr2

```

1080714 6.318761 1.0 ENSG00000205268.10 PDE7A chr8

```
[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_maleSpecific.txt',
      sep='\t', index=False, header=True)
shared_df
```

```
[20]:
```

	gene_id	ensemblID	seqname	gene_name	Dir
0	ENSG00000198286.9	ENSG00000198286	chr7	CARD11	-1.0
1	ENSG00000253988.1	ENSG00000253988	chr8	RP11-489018.1	-1.0

```
[21]: dlpfc.merge(gtf_annot[['gene_id', 'gene_name', 'seqname']],
      left_index=True, right_on='gene_id') \
      .merge(pd.DataFrame({'ensemblID': list(set(phase2_dlpfc['ensemblID']) &
      set(phase2_hippo['ensemblID']))}),
      on='ensemblID')
```

```
[21]:
```

	Feature	ensemblID	adj.P.Val	logFC	t	Dir	\
0	ENSG00000159958.5	ENSG00000159958	0.001774	-0.523950	-5.147635	-1.0	
1	ENSG00000198286.9	ENSG00000198286	0.004835	-0.236040	-4.585760	-1.0	
2	ENSG00000171659.13	ENSG00000171659	0.012267	-0.516121	-4.135553	-1.0	
3	ENSG00000253988.1	ENSG00000253988	0.018819	-0.555268	-3.943861	-1.0	
4	ENSG00000172243.17	ENSG00000172243	0.020343	-0.470366	-3.917891	-1.0	
5	ENSG00000184574.9	ENSG00000184574	0.024611	-0.371156	-3.807647	-1.0	
6	ENSG00000182578.13	ENSG00000182578	0.047930	-0.330906	-3.432998	-1.0	

	gene_id	gene_name	seqname
0	ENSG00000159958.5	TNFRSF13C	chr22
1	ENSG00000198286.9	CARD11	chr7
2	ENSG00000171659.13	GPR34	chrX
3	ENSG00000253988.1	RP11-489018.1	chr8
4	ENSG00000172243.17	CLEC7A	chr12
5	ENSG00000184574.9	LPAR5	chr12
6	ENSG00000182578.13	CSF1R	chr5

```
[22]: dlpfc.merge(gtf_annot[['gene_id', 'gene_name', 'seqname']],
      left_index=True, right_on='gene_id') \
      .merge(pd.DataFrame({'ensemblID': list(set(phase2_dlpfc['ensemblID']) &
      set(phase3_caudate['ensemblID']))}),
      on='ensemblID')
```



[22]:

	Feature	ensemblID	adj.P.Val	logFC	t	Dir	\
0	ENSG00000135697.9	ENSG00000135697	0.000161	0.567056	5.873402	1.0	
1	ENSG00000198286.9	ENSG00000198286	0.004835	-0.236040	-4.585760	-1.0	
2	ENSG00000095303.14	ENSG00000095303	0.005603	-0.340625	-4.514856	-1.0	
3	ENSG00000270095.1	ENSG00000270095	0.007770	0.181196	4.329651	1.0	
4	ENSG00000177990.11	ENSG00000177990	0.008681	0.160432	4.291095	1.0	
5	ENSG00000162747.9	ENSG00000162747	0.013554	0.710359	4.100150	1.0	
6	ENSG00000253988.1	ENSG00000253988	0.018819	-0.555268	-3.943861	-1.0	
7	ENSG00000168952.15	ENSG00000168952	0.022333	-0.100187	-3.858805	-1.0	
8	ENSG00000106714.17	ENSG00000106714	0.022511	0.111804	3.852368	1.0	
9	ENSG00000107719.8	ENSG00000107719	0.034600	-0.161925	-3.616218	-1.0	
10	ENSG00000164326.4	ENSG00000164326	0.035711	0.429425	3.587478	1.0	
11	ENSG00000178573.6	ENSG00000178573	0.040637	-0.129271	-3.509009	-1.0	
12	ENSG00000203734.11	ENSG00000203734	0.041004	0.327972	3.503061	1.0	
13	ENSG00000107099.15	ENSG00000107099	0.047995	-0.282705	-3.431753	-1.0	

	gene_id	gene_name	seqname
0	ENSG00000135697.9	BC01	chr16
1	ENSG00000198286.9	CARD11	chr7
2	ENSG00000095303.14	PTGS1	chr9
3	ENSG00000270095.1	RP11-214K3.18	chr12
4	ENSG00000177990.11	DPY19L2	chr12
5	ENSG00000162747.9	FCGR3B	chr1
6	ENSG00000253988.1	RP11-489O18.1	chr8
7	ENSG00000168952.15	STXBP6	chr14
8	ENSG00000106714.17	CNTNAP3	chr9
9	ENSG00000107719.8	PALD1	chr10
10	ENSG00000164326.4	CARTPT	chr5
11	ENSG00000178573.6	MAF	chr16
12	ENSG00000203734.11	ECT2L	chr6
13	ENSG00000107099.15	DOCK8	chr9

[23]:

```
hippo.merge(gtf_annot[['gene_id', 'gene_name', 'seqname']],
             left_index=True, right_on='gene_id')\
    .merge(pd.DataFrame({'ensemblID': list(set(phase2_hippo['ensemblID']) &
                                             set(phase3_caudate['ensemblID']))}),
           on='ensemblID')
```

[23]:

	Feature	ensemblID	adj.P.Val	logFC	t	Dir	\
0	ENSG00000157303.10	ENSG00000157303	0.000647	-0.682818	-5.735726	-1.0	
1	ENSG00000253988.1	ENSG00000253988	0.001467	-0.804170	-5.300010	-1.0	
2	ENSG00000110876.9	ENSG00000110876	0.002355	-0.497023	-5.086652	-1.0	
3	ENSG00000159618.15	ENSG00000159618	0.005969	-0.581969	-4.771118	-1.0	
4	ENSG00000140749.8	ENSG00000140749	0.007113	-0.589998	-4.682800	-1.0	
5	ENSG00000249740.2	ENSG00000249740	0.010842	0.567973	4.506132	1.0	
6	ENSG00000249738.8	ENSG00000249738	0.014382	-0.412936	-4.347224	-1.0	
7	ENSG00000009790.14	ENSG00000009790	0.017282	-0.436077	-4.258910	-1.0	

8	ENSG000000084734.8	ENSG000000084734	0.025109	0.336828	4.061594	1.0
9	ENSG000000198286.9	ENSG000000198286	0.029920	-0.456828	-3.984090	-1.0
10	ENSG000000053501.12	ENSG000000053501	0.031459	-0.120799	-3.953364	-1.0
11	ENSG000000270048.1	ENSG000000270048	0.031459	0.200951	3.949902	1.0
12	ENSG000000235750.9	ENSG000000235750	0.044530	0.483569	3.771899	1.0
13	ENSG000000104517.12	ENSG000000104517	0.046450	0.073471	3.729173	1.0

	gene_id	gene_name	seqname
0	ENSG000000157303.10	SUSD3	chr9
1	ENSG000000253988.1	RP11-489018.1	chr8
2	ENSG000000110876.9	SELPLG	chr12
3	ENSG000000159618.15	ADGRG5	chr16
4	ENSG000000140749.8	IGSF6	chr16
5	ENSG000000249740.2	OSMR-AS1	chr5
6	ENSG000000249738.8	AC008697.1	chr5
7	ENSG000000009790.14	TRAF3IP3	chr1
8	ENSG000000084734.8	GCKR	chr2
9	ENSG000000198286.9	CARD11	chr7
10	ENSG000000053501.12	USE1	chr19
11	ENSG000000270048.1	RP11-214K3.22	chr12
12	ENSG000000235750.9	KIAA0040	chr1
13	ENSG000000104517.12	UBR5	chr8

```
[24]: hippo.merge(gtf_annot[['gene_id', 'gene_name', 'seqname']],
                left_index=True, right_on='gene_id')\
                .merge(pd.DataFrame({'ensemblID':
→list(set(phase2_dlpfc['ensemblID']) &
→set(phase2_hippo['ensemblID']))}),
                    on='ensemblID')
```

[24]:	Feature	ensemblID	adj.P.Val	logFC	t	Dir	\
0	ENSG000000253988.1	ENSG000000253988	0.001467	-0.804170	-5.300010	-1.0	
1	ENSG000000159958.5	ENSG000000159958	0.002355	-0.572210	-5.111428	-1.0	
2	ENSG000000172243.17	ENSG000000172243	0.024160	-0.520501	-4.114291	-1.0	
3	ENSG000000171659.13	ENSG000000171659	0.025109	-0.508756	-4.061452	-1.0	
4	ENSG000000198286.9	ENSG000000198286	0.029920	-0.456828	-3.984090	-1.0	
5	ENSG000000184574.9	ENSG000000184574	0.031833	-0.393921	-3.940339	-1.0	
6	ENSG000000182578.13	ENSG000000182578	0.038886	-0.366242	-3.856809	-1.0	

	gene_id	gene_name	seqname
0	ENSG000000253988.1	RP11-489018.1	chr8
1	ENSG000000159958.5	TNFRSF13C	chr22
2	ENSG000000172243.17	CLEC7A	chr12
3	ENSG000000171659.13	GPR34	chrX
4	ENSG000000198286.9	CARD11	chr7
5	ENSG000000184574.9	LPAR5	chr12

6 ENSG00000182578.13 CSF1R chr5

```
[25]: dd = np.sum(shared_df.seqname.isin(['chrX', 'chrY'])) / shared_df.shape[0] * 100
print("%.2f%% of shared DEG are allosomal!" % dd)
```

0.00% of shared DEG are allosomal!

```
[26]: 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_maleSpecific.csv')
```

## 1.2 Comparison with CommonMind

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

```
[27]: Dir
-1.0    63
 1.0   109
dtype: int64
```

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

```
[28]: (172, 6)
```

### 1.2.1 Upset Plot

```
[29]: 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()
```

```
[30]: 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)
```

```
(2211, 2) (2211, 2) (2211, 2) (2211, 2)
```

```
[31]: 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)
```

```
[31]:
```

	Caudate	DLPFC	Hippocampus	CMC DLPFC
ensemblID				
ENSG00000000971	1	0	0	0
ENSG00000002330	1	0	0	0

```
[32]: %R
cmc = subset_pvalue('../.../cmc_dlpfc/male_analysis/metrics_summary/_m/
↳male_specific_DE_genes.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, 2000),
  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, 2000),
  gp=gpar(fill=cbb_palette[comb_degree(m)]),
  annotation_name_rot = 90)

pdf('cmc_sex_tissue_upsetR_DEgenes_maleSpecific.pdf', width=10, 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\ncsize" = 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, 2000),
    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_maleSpecific.pdf", width=6, ↪
  ↪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\ncsize", {
  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/male_analysis/metrics_summary/_m/
↳male_specific_DE_genes.txt',
                        sep='\t')
dft['Dir'] = np.sign(dft['t'])
dft.head()
```

```
[35]:
```

	Feature	gencodeID	Symbol	ensemblID	Chrom	\
0	ENSG00000119411.10	ENSG00000119411.10	BSPRY	ENSG00000119411	chr9	
1	ENSG00000159871.14	ENSG00000159871.14	LYPD5	ENSG00000159871	chr19	
2	ENSG00000231752.5	ENSG00000231752.5	EMBP1	ENSG00000231752	chr1	
3	ENSG00000163833.7	ENSG00000163833.7	FBXO40	ENSG00000163833	chr3	
4	ENSG00000158457.5	ENSG00000158457.5	TSPAN33	ENSG00000158457	chr7	

	logFC	t	adj.P.Val	Female_Pval	Female_FDR	Type	Dir
0	0.280110	5.861029	0.000028	0.439030	0.454209	gene	1.0
1	0.213822	5.885126	0.000028	0.362537	0.391824	gene	1.0
2	-0.189426	-5.536119	0.000083	0.085029	0.116914	gene	-1.0
3	0.253446	5.285296	0.000219	0.232535	0.268937	gene	1.0
4	0.167825	4.670299	0.001520	0.214432	0.252556	gene	1.0

```
[36]: shared_df = dft.loc[:, ['Feature', 'ensemblID', 'Chrom', 'Symbol', '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.to_csv('cmc_shared_degs_annotation_maleSpecific.txt', sep='\t',
                index=False, header=True)
shared_df
```

```
[36]: Empty DataFrame
Columns: [Feature, ensemblID, Chrom, Symbol, Dir]
Index: []
```

```
[37]: cmc.merge(gtf_annot[['gene_id', 'ensemblID', 'gene_name', 'seqname']],
              on='ensemblID') \
      .merge(pd.DataFrame({'ensemblID': list(set(phase2_dlpfc['ensemblID']) &
              set(cmc['ensemblID']))}),
              on='ensemblID')
```

```
[37]:
```

	ensemblID	CMC	DLPFC	gene_id	gene_name	seqname
0	ENSG000000156414		1	ENSG000000156414.18	TDRD9	chr14
1	ENSG000000171488		1	ENSG000000171488.14	LRRC8C	chr1
2	ENSG000000231752		1	ENSG000000231752.5	EMBP1	chr1

```
[38]: cmc.merge(gtf_annot[['gene_id', 'ensemblID', 'gene_name', 'seqname']],
               on='ensemblID')\
      .merge(pd.DataFrame({'ensemblID': list(set(phase2_hippo['ensemblID']) &
                                             set(cmc['ensemblID']))}),
               on='ensemblID')
```

```
[38]: Empty DataFrame
Columns: [ensemblID, CMC DLPFC, gene_id, gene_name, seqname]
Index: []
```

```
[39]: cmc.merge(gtf_annot[['gene_id', 'ensemblID', 'gene_name', 'seqname']],
               on='ensemblID')\
      .merge(pd.DataFrame({'ensemblID': list(set(phase3_caudate['ensemblID']) &
                                             set(cmc['ensemblID']))}),
               on='ensemblID')
```

```
[39]:
```

	ensemblID	CMC	DLPFC	gene_id	gene_name	seqname
0	ENSG000000066185		1	ENSG000000066185.12	ZMYND12	chr1
1	ENSG000000100116		1	ENSG000000100116.16	GCAT	chr22
2	ENSG000000100266		1	ENSG000000100266.18	PACSIN2	chr22
3	ENSG000000115170		1	ENSG000000115170.13	ACVR1	chr2
4	ENSG000000134597		1	ENSG000000134597.14	RBMX2	chrX
5	ENSG000000139372		1	ENSG000000139372.14	TDG	chr12
6	ENSG000000189410		1	ENSG000000189410.11	SH2D5	chr1
7	ENSG000000256463		1	ENSG000000256463.8	SALL3	chr18
8	ENSG000000260400		1	ENSG000000260400.1	RP11-119F7.5	chr10

```
[40]: 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_maleSpecific.csv')
```

```
[ ]:
```