main

August 16, 2021

1 Extract unique female specific SZ-associated genes

[1]:

```
import functools
import numpy as np
import pandas as pd
from os import environ
from gtfparse import read_gtf
from scipy.stats import mannwhitneyu
from statsmodels.stats.multitest import fdrcorrection
```

[2]:

```
environ['NUMEXPR_MAX_THREADS'] = '16'
```

1.1 Functions

1.1.1 Cached functions

[3]:

```
@functools.lru_cache()
def get_res_df(feature):
    return pd.read_csv('../../../../../interaction_model/caudate/_m/%s/
 ↪residualized_expression.tsv' %
                       feature, sep='\t').T


@functools.lru_cache()
def get_pheno_df():
    return pd.read_csv('/ceph/projects/v3_phase3_paper/inputs/phenotypes/_m/
 ↪caudate_phenotypes.csv',
                       index_col=0)


@functools.lru_cache()
def get_res_pheno_df(feature):
    return pd.merge(get_pheno_df(), get_res_df(feature), left_index=True,␣
 ↪right_index=True)


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

1.1.2 Simple functions

```
[4]: def gene_annot(feature):
    gtf_file = '/ceph/genome/human/gencode25/gtf.CHR/_m/gencode.v25.annotation.
 ↪gtf'
    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"]]


def get_de(feature):
    f = pd.read_csv('../../../female_analysis/_m/%s/diffExpr_szVctl_full.txt' %␣
 ↪feature,
                    sep='\t', index_col=0)\
          .rename(columns={'gencodeGeneID': 'gencodeID'})
    m = pd.read_csv('../../../male_analysis/_m/%s/diffExpr_szVctl_full.txt' %␣
 ↪feature,
                    sep='\t', index_col=0)\
          .rename(columns={'gencodeGeneID': 'gencodeID'})
    a = pd.read_csv('/ceph/projects/v4_phase3_paper/analysis/
 ↪differential_expression/'+\
                    '_m/%s/diffExpr_szVctl_full.txt' % feature,
                    sep='\t', index_col=0)\
          .rename(columns={'gencodeGeneID': 'gencodeID'})
    return f, m, a


def get_unique(x, y, thres=0.05):
    return x.merge(pd.DataFrame(index = list(set(x[(x['adj.P.Val'] <= thres)].
 ↪index) -
                                             set(y[(y['adj.P.Val'] <= thres)].
 ↪index))),
                   left_index=True, right_index=True)


def subset_sz_male(feature):
    df = get_res_pheno_df(feature)
    ctl = df[(df['Dx'] == 'Control') & (df['Sex'] == 'M')].copy()
    sz = df[(df['Dx'] == 'Schizo') & (df['Sex'] == 'M')].copy()
    return ctl, sz


def add_pvals_adjustPval(feature, df):
    ctl, sz = subset_sz_male(feature)
    pval_df = []
    for gene_id in df.Feature:
        stat, pval = mannwhitneyu(ctl[gene_id], sz[gene_id])
        pval_df.append(pval)
    fdr_df = fdrcorrection(pval_df)
    return pd.concat([df.set_index('Feature'),
                      pd.DataFrame({'Male_Pval': pval_df, 'Male_FDR':␣
 ↪fdr_df[1]},
                                   index=df.Feature)], axis=1)
```

1.2 Genes

[5]:
```
gtf_annot = gene_annot('gene')
f, m, a = get_de('genes')
f['Feature'] = f.index
```

```
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']
```

[6]:
```
#genes = get_unique(get_unique(f, m), a)
genes = get_unique(f, m)
genes = pd.merge(gtf_annot[['gene_id', 'seqname']], genes, left_on='gene_id',
                 right_on='Feature', how='right').rename(columns={'seqname':␣
 ↪'Chrom'})
genes = genes[['Feature', 'gencodeID', 'Symbol', 'ensemblID',
               'Chrom', 'logFC', 't', 'adj.P.Val']].sort_values('adj.P.Val')
genes = add_pvals_adjustPval('genes', genes)
genes = genes[~(genes['Male_Pval'] < 0.05)].sort_values('adj.P.Val').
 ↪reset_index() ## Stringents
genes['Type'] = 'gene'
genes.head(2)
```

[6]:
```
              Feature            gencodeID   Symbol        ensemblID  Chrom  \
0   ENSG00000070915.9   ENSG00000070915.9  SLC12A3  ENSG00000070915  chr16
1  ENSG00000111181.12  ENSG00000111181.12  SLC6A12  ENSG00000111181  chr12

      logFC         t  adj.P.Val  Male_Pval  Male_FDR  Type
0  0.909953  4.668428   0.006380   0.349666  0.368739  gene
1 -0.432628 -4.405278   0.009994   0.086899  0.136219  gene
```

1.3 Transcripts

[7]:
```
gtf_annot = gene_annot('transcript')

f, m, a = get_de('transcripts')
```

```
f['Feature'] = f.index
f['ensemblID'] = f.gene_id.str.replace('\\.\d+', '', regex=True)
#trans = get_unique(get_unique(f, m), a)
trans = get_unique(f, m)
trans = pd.merge(gtf_annot[['transcript_id', 'seqname']], trans,
                 left_on='transcript_id', right_on='Feature',
                 how='right').rename(columns={'seqname': 'Chrom'}).
 ↪sort_values('adj.P.Val')
trans = trans[['Feature', 'gene_id', 'gene_name', 'ensemblID', 'Chrom',
               'logFC', 't', 'adj.P.Val']].rename(columns={'gene_name':␣
 ↪'Symbol',
                                                            'gene_id':␣
 ↪'gencodeID'})
trans = add_pvals_adjustPval('transcripts', trans)
trans = trans[~(trans['Male_Pval'] <= 0.05)].sort_values('adj.P.Val').
 ↪reset_index() ## Stringents
trans['Type'] = 'transcript'
trans.head(2)
```

[7]:

```
             Feature           gencodeID Symbol        ensemblID  Chrom  \
0  ENST00000546602.5  ENSG00000111077.17   TNS2  ENSG00000111077  chr12
1  ENST00000382297.6  ENSG00000107104.18  KANK1  ENSG00000107104   chr9

      logFC         t  adj.P.Val  Male_Pval  Male_FDR        Type
0 -2.895591 -4.732890   0.019794   0.369834  0.381391  transcript
1 -0.475026 -4.463686   0.036958   0.066030  0.087160  transcript
```

1.3.1 Exons

[8]:

```
gtf_annot = gene_annot('exon')
gtf_annot['ensemblID'] = gtf_annot.gene_id.str.replace('\\.\d+', '', regex=True)

f, m, a = get_de('exons')
f['Feature'] = f.index
#exons = get_unique(get_unique(f, m), a)
exons = get_unique(f, m)
exons = pd.merge(gtf_annot[['ensemblID', 'seqname']], exons,
                 on='ensemblID', how='right').rename(columns={'seqname':␣
 ↪'Chrom'})
exons = exons[['Feature', 'gencodeID', 'Symbol', 'ensemblID',
               'Chrom', 'logFC', 't', 'adj.P.Val']].groupby('Feature')\
            .first().reset_index().sort_values('adj.P.Val')
exons = add_pvals_adjustPval('exons', exons)
exons = exons[~(exons['Male_Pval'] <= 0.05)].sort_values('adj.P.Val').
 ↪reset_index()
exons['Type'] = 'exon'
exons.head(2)
```

```
[8]:    Feature           gencodeID  Symbol        ensemblID Chrom     logFC  \
     0   e36382  ENSG00000085831.15  TTC39A  ENSG00000085831  chr1  0.666415
     1  e193774  ENSG00000077044.10    DGKD  ENSG00000077044  chr2  0.253319

               t  adj.P.Val  Male_Pval  Male_FDR  Type
     0  4.866389   0.024742   0.065018  0.088508  exon
     1  4.830409   0.026066   0.118827  0.137924  exon
```

1.3.2 Junctions

```
[9]: f, m, a = get_de('junctions')
     f['Feature'] = f.index
     #juncs = get_unique(get_unique(f, m), a)
     juncs = get_unique(f, m)
     juncs = pd.merge(gtf_annot[['ensemblID', 'seqname']], juncs,
                      on='ensemblID', how='right').rename(columns={'seqname':␣
      ↪'Chrom'})
     juncs = juncs[['Feature', 'gencodeID', 'Symbol', 'ensemblID', 'Chrom',
                    'logFC', 't', 'adj.P.Val']].groupby('Feature')\
             .first().reset_index().sort_values('adj.P.Val')
     juncs = add_pvals_adjustPval('junctions', juncs)
     juncs = juncs[~(juncs['Male_Pval'] <= 0.05)].sort_values('adj.P.Val').
      ↪reset_index() ## Stringents
     juncs['Type'] = 'junction'
     juncs.head(2)
```

```
[9]:                     Feature           gencodeID Symbol        ensemblID  \
     0  chr11:65505844-65505919(+)                None   None             None
     1      chr11:502250-504823(-)  ENSG00000023191.16   RNH1  ENSG00000023191

        Chrom     logFC         t  adj.P.Val  Male_Pval  Male_FDR      Type
     0   None  2.410128  4.839635   0.017846   0.386719  0.392578  junction
     1  chr11 -0.505327 -4.812657   0.018865   0.077289  0.097704  junction
```

1.4 DE summary

1.4.1 DE (feature)

```
[10]: gg = len(set(genes['Feature']))
      tt = len(set(trans['Feature']))
      ee = len(set(exons['Feature']))
      jj = len(set(juncs['Feature']))

      print("\nGene:\t\t%d\nTranscript:\t%d\nExon:\t\t%d\nJunction:\t%d" %
            (gg, tt, ee, jj))
```

```
Gene:           30
```

```
Transcript:     11
Exon:           19
Junction:       17
```

DE (EnsemblID)

[11]:
```
gg = len(set(genes['ensemblID']))
tt = len(set(trans['ensemblID']))
ee = len(set(exons['ensemblID']))
jj = len(set(juncs['ensemblID']))

print("\nGene:\t\t%d\nTranscript:\t%d\nExon:\t\t%d\nJunction:\t%d" %
      (gg, tt, ee, jj))
```

```
Gene:           30
Transcript:     11
Exon:           15
Junction:       15
```

DE (Gene Symbol)

[12]:
```
gg = len(set(genes['Symbol']))
tt = len(set(trans['Symbol']))
ee = len(set(exons['Symbol']))
jj = len(set(juncs['Symbol']))

print("\nGene:\t\t%d\nTranscript:\t%d\nExon:\t\t%d\nJunction:\t%d" % (gg, tt,␣
 ↪ee, jj))
```

```
Gene:           26
Transcript:     11
Exon:           15
Junction:       15
```

1.4.2 Feature effect size summary

[13]:
```
feature_list = ['Genes', 'Transcript', 'Exons', 'Junctions']
feature_df = [genes, trans, exons, juncs]
for ii in range(4):
    ff = feature_df[ii]
    half = len(set(ff[(np.abs(ff['logFC']) >= 0.5)].Feature))
    one = len(set(ff[(np.abs(ff['logFC']) >= 1)].Feature))
    print("\nThere are %d unique %s with abs(log2FC) >= 0.5" % (half,␣
 ↪feature_list[ii]))
    print("There are %d unique %s with abs(log2FC) >= 1" % (one,␣
 ↪feature_list[ii]))
```

```
There are 12 unique Genes with abs(log2FC) >= 0.5
There are 1 unique Genes with abs(log2FC) >= 1

There are 6 unique Transcript with abs(log2FC) >= 0.5
There are 1 unique Transcript with abs(log2FC) >= 1

There are 7 unique Exons with abs(log2FC) >= 0.5
There are 0 unique Exons with abs(log2FC) >= 1

There are 10 unique Junctions with abs(log2FC) >= 0.5
There are 1 unique Junctions with abs(log2FC) >= 1
```

[14]:
```
feature_list = ['Genes', 'Transcripts', 'Exons', 'Junctions']
feature_df = [genes, trans, exons, juncs]
for ii in range(4):
    ff = feature_df[ii]
    half = len(set(ff[(np.abs(ff['logFC']) >= 0.5)].ensemblID))
    one = len(set(ff[(np.abs(ff['logFC']) >= 1)].ensemblID))
    print("\nThere are %d unique %s with abs(log2FC) >= 0.5" % (half,␣
 ↪feature_list[ii]))
    print("There are %d unique %s with abs(log2FC) >= 1" % (one,␣
 ↪feature_list[ii]))
```

```
There are 12 unique Genes with abs(log2FC) >= 0.5
There are 1 unique Genes with abs(log2FC) >= 1

There are 6 unique Transcripts with abs(log2FC) >= 0.5
There are 1 unique Transcripts with abs(log2FC) >= 1

There are 6 unique Exons with abs(log2FC) >= 0.5
There are 0 unique Exons with abs(log2FC) >= 1

There are 9 unique Junctions with abs(log2FC) >= 0.5
There are 1 unique Junctions with abs(log2FC) >= 1
```

[15]:
```
df = pd.concat([genes, trans, exons, juncs], axis=0)
df.to_csv('female_specific_DE_4features.txt', sep='\t', index=False,␣
 ↪header=True)
```

1.5 Number of DEGs on allosomes

[16]:
```
df[(df['Chrom'].isin(['chrX', 'chrY']))].groupby(['Type', 'Chrom']).size()
```

[16]:
```
Type  Chrom
gene  chrX     2
dtype: int64
```

[ ]:

```

```