main

July 14, 2021

1 Extract unique male 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'
```

[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)
```

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


def add_pvals_adjustPval(feature, df):
    ctl, sz = subset_sz_female(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({'Female_Pval': pval_df,
                                    'Female_FDR': fdr_df[1]}, index=df.
 ↪Feature)],
                     axis=1)
```

1.1 Genes

[5]:

```
gtf_annot = gene_annot('gene')

f, m, a = get_de('genes')
m['Feature'] = m.index
#genes = get_unique(get_unique(m, f), a)
genes = get_unique(m, f)
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['Female_Pval'] <= 0.05)].sort_values('adj.P.Val') ##␣
 ↪Stringents
genes['Type'] = 'gene'
genes.shape
```

```
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']
```

[5]:

```
(1858, 10)
```

1.2 Transcripts

[6]:

```
gtf_annot = gene_annot('transcript')

f, m, a = get_de('transcripts')
m['Feature'] = m.index
m['ensemblID'] = m.gene_id.str.replace('\\.\d+', '', regex=True)
#trans = get_unique(get_unique(m, f), a)
trans = get_unique(m, f)
trans = pd.merge(gtf_annot[['transcript_id', 'seqname']], trans,
                 on='transcript_id', how='right').rename(columns={'seqname':␣
 ↪'Chrom'})
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['Female_Pval'] <= 0.05)].sort_values('adj.P.Val') ##␣
 ↪Stringents
trans['Type'] = 'transcript'
trans.shape
```

[6]:
```
(480, 10)
```

1.2.1 Exons

[7]:
```
gtf_annot = gene_annot('exon')
gtf_annot['ensemblID'] = gtf_annot.gene_id.str.replace('\\.\d+', '', regex=True)
```

[8]:
```
f, m, a = get_de('exons')
m['Feature'] = m.index
#exons = get_unique(get_unique(m, f), a)
exons = get_unique(m, f)
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']].sort_values('adj.P.Val')\
        .groupby('Feature').first().reset_index()
exons = add_pvals_adjustPval('exons', exons)
exons = exons[~(exons['Female_Pval'] <= 0.05)].sort_values('adj.P.Val') ##␣
 ↪Stringents
exons['Type'] = 'exon'
exons.shape
```

[8]:
```
(5834, 10)
```

1.2.2 Junctions

[9]:
```
f, m, a = get_de('junctions')
m['Feature'] = m.index
#juncs = get_unique(get_unique(m, f), a)
juncs = get_unique(m, f)
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']].sort_values('adj.P.Val')\
        .groupby('Feature').first().reset_index()
```

```
juncs = add_pvals_adjustPval('junctions', juncs)
juncs['Type'] = 'junction'
juncs = juncs[~(juncs['Female_Pval'] <= 0.05)].sort_values('adj.P.Val') ##␣
 ↪Stringents
juncs.shape
```

[9]:
```
(2064, 10)
```

1.3 DE summary

1.3.1 DE (feature)

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

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

```
Gene:           1858
Transcript:     480
Exon:           5834
Junction:       2064
```

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:           1858
Transcript:     460
Exon:           1830
Junction:       1062
```

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:           1675
Transcript:     459
Exon:           1761
Junction:       1062
```

1.3.2 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)].index))
    one = len(set(ff[(np.abs(ff['logFC']) >= 1)].index))
    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 14 unique Genes with abs(log2FC) >= 0.5
There are 0 unique Genes with abs(log2FC) >= 1

There are 79 unique Transcript with abs(log2FC) >= 0.5
There are 28 unique Transcript with abs(log2FC) >= 1

There are 121 unique Exons with abs(log2FC) >= 0.5
There are 1 unique Exons with abs(log2FC) >= 1

There are 67 unique Junctions with abs(log2FC) >= 0.5
There are 5 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 14 unique Genes with abs(log2FC) >= 0.5
There are 0 unique Genes with abs(log2FC) >= 1
There are 78 unique Transcripts with abs(log2FC) >= 0.5
There are 27 unique Transcripts with abs(log2FC) >= 1

There are 35 unique Exons with abs(log2FC) >= 0.5
There are 1 unique Exons with abs(log2FC) >= 1

There are 25 unique Junctions with abs(log2FC) >= 0.5
There are 1 unique Junctions with abs(log2FC) >= 1
```

```
[15]: df = pd.concat([genes.reset_index(), trans.reset_index(),
                      exons.reset_index(), juncs.reset_index()], axis=0)
      df.to_csv('male_specific_DE_4features.txt', sep='\t', index=False, header=True)
```

1.4 Number of DEGs on allosomes

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

```
[16]: Type        Chrom
      exon        chrX     202
                  chrY       6
      gene        chrX      60
                  chrY       4
      junction    chrX      71
      transcript  chrX       5
                  chrY       2
      dtype: int64
```

```
[ ]:
```