main

August 12, 2021

1 Summary of eQTL analysis

[1]:
```
import functools
import pandas as pd
```

[2]:
```
config = {
    "genes": "/ceph/projects/v4_phase3_paper/inputs/counts/text_files_counts/_m/
 ↪caudate/gene_annotation.tsv",
    "transcripts": "/ceph/projects/v4_phase3_paper/inputs/counts/
 ↪text_files_counts/_m/caudate/tx_annotation.tsv",
    "exons": "/ceph/projects/v4_phase3_paper/inputs/counts/text_files_counts/_m/
 ↪caudate/exon_annotation.tsv",
    "junctions": "/ceph/projects/v4_phase3_paper/inputs/counts/
 ↪text_files_counts/_m/caudate/jxn_annotation.tsv"
}
```

1.1 Functions

[3]:
```
@functools.lru_cache()
def get_eqtls(feature):
    fn = "/ceph/projects/v4_phase3_paper/analysis/eqtl_analysis/all/%s/
 ↪expression_gct/" % feature +\
         "prepare_expression/annotate_outputs/_m/Brainseq_LIBD.signifpairs.txt.
 ↪gz"
    return pd.read_csv(fn, sep='\t')


@functools.lru_cache()
def annotate_eqtls(feature):
    annot = pd.read_csv(config[feature], sep='\t').loc[:, ["names",␣
 ↪"gencodeID"]]
    return get_eqtls(feature).merge(annot, left_on="gene_id", right_on="names").
 ↪drop(["names"], axis=1)


@functools.lru_cache()
def load_pgc2():
    pgc2_file = '/ceph/projects/v4_phase3_paper/inputs/sz_gwas/'+\
                'pgc2_clozuk/map_phase3/_m/libd_hg38_pgc2sz_snps_p5e_minus8.tsv'
    return pd.read_csv(pgc2_file, sep='\t', low_memory=False)


@functools.lru_cache()
def merge_pgc2_N_eqtl(feature):
    return load_pgc2().merge(annotate_eqtls(feature), how='inner',
                             left_on='our_snp_id', right_on='variant_id',
                             suffixes=['_PGC2', '_eqtl'])
```

1.2 Load data

1.2.1 Load significant eQTLs after permutation analysis

[4]:

```
genes = annotate_eqtls("genes")
trans = annotate_eqtls("transcripts")
exons = annotate_eqtls("exons")
juncs = annotate_eqtls("junctions")
```

1.2.2 Load PGC2+CLOZUK annotated eQTLs

[5]:

```
genes2 = merge_pgc2_N_eqtl("genes")
trans2 = merge_pgc2_N_eqtl("transcripts")
exons2 = merge_pgc2_N_eqtl("exons")
juncs2 = merge_pgc2_N_eqtl("junctions")
```

1.3 Summarize results eQTL analysis

1.3.1 Total significant gene-variant pairs

[6]:

```
gg = genes.shape[0]
tt = trans.shape[0]
ee = exons.shape[0]
jj = juncs.shape[0]

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

```

Gene:           2242055
Transcript:     3041906
Exon:           4783603
Junction:       5052809
```

1.3.2 Total significant eGenes

[7]:

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

print("\neGene:\t\t%d\neTranscript:\t%d\neExon:\t\t%d\neJunction:\t%d" %
      (gg, tt, ee, jj))
```

```
eGene:          16014
eTranscript:    26092
eExon:          42510
eJunction:      46804
```

1.3.3 Total significant eGenes

[8]:

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

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

```
Gene:           16014
Transcript:     13700
Exon:           13910
Junction:       10087
```

1.4 Summarize results eQTL analysis overlapping with PGC2+CLOZUK SNPs

1.4.1 Total significant gene-variant pairs

[9]:

```
gg = genes2.shape[0]
tt = trans2.shape[0]
ee = exons2.shape[0]
jj = juncs2.shape[0]

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

```
Gene:           40139
Transcript:     60356
Exon:           75669
Junction:       98419
```

1.4.2 Total significant eGenes

[10]:

```
gg = len(set(genes2['gene_id']))
tt = len(set(trans2['gene_id']))
ee = len(set(exons2['gene_id']))
jj = len(set(juncs2['gene_id']))

print("\neGene:\t\t%d\neTranscript:\t%d\neExon:\t\t%d\neJunction:\t%d" %
      (gg, tt, ee, jj))
```

```
eGene:          382
eTranscript:    576
eExon:          855
eJunction:      937
```

1.4.3 Total significant eFeatures

[11]:

```
gg = len(set(genes2['gencodeID']))
tt = len(set(trans2['gencodeID']))
ee = len(set(exons2['gencodeID']))
jj = len(set(juncs2['gencodeID']))

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

```
Gene:           382
Transcript:     342
Exon:           337
Junction:       255
```

1.5 Save significant results

1.5.1 All associations

[12]:

```
genes["Type"] = "Gene"
trans["Type"] = "Transcript"
exons["Type"] = "Exon"
juncs["Type"] = "Junction"

df = pd.concat([genes, trans, exons, juncs])\
       .loc[:, ["variant_id", "gene_id", "gencodeID", "tss_distance",␣
 ↪"ma_samples", "ma_count",
                "maf", "slope", "slope_se", "pval_nominal",␣
 ↪"pval_nominal_threshold",
                "min_pval_nominal", "pval_beta", "Type"]]
df["Type"] = df.Type.astype("category").cat.reorder_categories(["Gene",␣
 ↪"Transcript", "Exon", "Junction"])
```

```
df.sort_values(["Type", "gene_id", "pval_nominal"])\
  .to_csv("Brainseq_LIBD_caudate_4features.signifpairs.txt.gz", sep='\t',␣
 ↪index=False)
```

1.5.2 PGC2+CLOZUK associated variants

[13]:

```
genes2["Type"] = "Gene"
trans2["Type"] = "Transcript"
exons2["Type"] = "Exon"
juncs2["Type"] = "Junction"

df = pd.concat([genes2, trans2, exons2, juncs2])\
       .loc[:, ["variant_id", "rsid", "hg38chrc", "gene_id", "gencodeID",␣
 ↪"maf", "Freq.A1", "A1",
                "slope", "slope_se", "OR", "SE", "P", "pval_nominal",␣
 ↪"pval_nominal_threshold",
                "pgc2_a1_same_as_our_counted", "is_index_snp", "Type"]]
df["Type"] = df.Type.astype("category").cat.reorder_categories(["Gene",␣
 ↪"Transcript", "Exon", "Junction"])
df.sort_values(["Type", "gene_id", "pval_nominal", "P"])\
  .to_csv("Brainseq_LIBD_caudate_4features_PGC2.signifpairs.txt.gz", sep='\t',␣
 ↪index=False)
```

[ ]:

```

```