main

August 31, 2021

1 Summary of interacting cis-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_mashr_eqtls(feature):
    df = pd.read_csv("../../_m/%s/significant_geneSNP_pairs_3tissues.tsv" %␣
 ↪feature, sep='\t')
    return df[(df["N_Regions_Shared"] == 1) & (df["Caudate"] == 1)]


@functools.lru_cache()
def annotate_eqtls(feature):
    annot = pd.read_csv(config[feature], sep='\t').loc[:, ["names",␣
 ↪"gencodeID"]]
    return get_mashr_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, index_col=0)


@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 caudate specific cis-eQTL, mashr

1.3.1 Total significant eGenes

[6]:
```
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:          1932
eTranscript:    6951
eExon:          6422
eJunction:      10369
```

1.3.2 Total significant eGenes

[7]:
```
gg = len(set(genes['gencodeID']))
tt = len(set(trans['gencodeID']))
ee = len(set(exons['gencodeID']))
```

```python
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:           1932
Transcript:     5371
Exon:           3528
Junction:       4931
```

1.4 Summarize results eQTL analysis overlapping with PGC2+CLOZUK SNPs

1.4.1 Total significant eGenes

[8]:

```python
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:          11
eTranscript:    32
eExon:          41
eJunction:      56
```

1.4.2 Total significant eFeatures

[9]:

```python
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:           11
Transcript:     30
Exon:           15
Junction:       31
```

1.5 Save significant results

1.5.1 All associations

[10]:
```
genes["Type"] = "Gene"
trans["Type"] = "Transcript"
exons["Type"] = "Exon"
juncs["Type"] = "Junction"

df = pd.concat([genes, trans, exons, juncs])
df["Type"] = df.Type.astype("category").cat.reorder_categories(["Gene",␣
 ↪"Transcript", "Exon", "Junction"])
df.sort_values(["Type", "gene_id"])\
  .to_csv("Brainseq_LIBD_caudate_specific_4features.eGenes.txt.gz", sep='\t',␣
 ↪index=False)
```

1.5.2 PGC2+CLOZUK associated variants

[11]:
```
genes2["Type"] = "Gene"
trans2["Type"] = "Transcript"
exons2["Type"] = "Exon"
juncs2["Type"] = "Junction"

df = pd.concat([genes2, trans2, exons2, juncs2])
df["Type"] = df.Type.astype("category").cat.reorder_categories(["Gene",␣
 ↪"Transcript", "Exon", "Junction"])
df.sort_values(["Type", "gene_id", "P"])\
  .to_csv("Brainseq_LIBD_caudate_specific_4features_PGC2.eGenes.txt.gz",␣
 ↪sep='\t', index=False)
```

[ ]:
```

```