main

September 27, 2021

1 Summary of interacting cis-eQTL analysis

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

1.1 Functions

1.1.1 Cached functions

[2]:
```
@functools.lru_cache()
def get_mashr_degs(feature, tissue):
    df = pd.read_csv("../../_m/%s/lfsr_feature_4tissues.txt.gz" % feature,
                     sep='\t').loc[:, ["Effect", tissue]]
    return df[(df[tissue] < 0.05)].rename(columns={tissue: "lfsr"})


@functools.lru_cache()
def annotate_degs(feature, tissue):
    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",
    }
    annot = pd.read_csv(config[feature], sep='\t').loc[:, ["names", "seqnames",␣
 ↪"gencodeID"]]
    return get_mashr_degs(feature, tissue).merge(annot, left_on="Effect",
                                                 right_on="names").
 ↪drop(["names"], axis=1)
```

1.1.2 Simple functions

[3]:
```python
def extract_features(tissue):
    ## Extract significant eQTL using mashr
    genes = annotate_degs("genes", tissue)
    trans = annotate_degs("transcripts", tissue)
    exons = annotate_degs("exons", tissue)
    juncs = annotate_degs("junctions", tissue)
    return genes, trans, exons, juncs


def output_summary(tissue, variable):
    ## Extract eQTL using mashr
    genes, trans, exons, juncs = extract_features(tissue)
    ## Total significant eQTLs
    gg = len(set(genes[variable]))
    tt = len(set(trans[variable]))
    ee = len(set(exons[variable]))
    jj = len(set(juncs[variable]))
    print("\nGene:\t\t%d\nTranscript:\t%d\nExon:\t\t%d\nJunction:\t%d" %
          (gg, tt, ee, jj))


def get_DEGs_result_by_tissue(tissue):
    genes, trans, exons, juncs = extract_features(tissue)
    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["Tissue"] = tissue.replace(".", " ")
    return df
```

1.2 Feature Summary

1.2.1 Summarize results mashr (local false sign rate < 0.05)

[4]:
```python
for tissue in ["Caudate", "Dentate.Gyrus", "DLPFC", "Hippocampus"]:
    print("")
    print(tissue)
    ## significant Features
    print("\nFeatures")
    output_summary(tissue, "Effect")
    print("\nGeneid")
    ## significant Geneid
    output_summary(tissue, "gencodeID")
```

```
Caudate

Features

Gene:           4979
Transcript:     8951
Exon:           38057
Junction:       11841

Geneid

Gene:           4979
Transcript:     6071
Exon:           10195
Junction:       4722

Dentate.Gyrus

Features

Gene:           4170
Transcript:     10578
Exon:           31058
Junction:       10435

Geneid

Gene:           4170
Transcript:     6971
Exon:           9393
Junction:       4635

DLPFC

Features

Gene:           4960
Transcript:     10229
Exon:           39318
Junction:       12404

Geneid

Gene:           4960
Transcript:     6787
Exon:           10529
Junction:       5087
```

```
Hippocampus

Features

Gene:           5029
Transcript:     10759
Exon:           38193
Junction:       11954

Geneid

Gene:           5029
Transcript:     6945
Exon:           10312
Junction:       4868
```

[5]:
```
caud8 = get_DEGs_result_by_tissue("Caudate")
gyrus = get_DEGs_result_by_tissue("Dentate.Gyrus")
dlpfc = get_DEGs_result_by_tissue("DLPFC")
hippo = get_DEGs_result_by_tissue("Hippocampus")
```

1.3 Save significant results

[6]:
```
pd.concat([caud8, gyrus, dlpfc, hippo])\
  .sort_values(["Tissue", "Type", "lfsr"])\
  .loc[:, ["Tissue", "Effect", "gencodeID", "seqnames", "lfsr", "Type"]]\
  .to_csv("BrainSeq_ancestry_4features_4regions.txt.gz", sep='\t', index=False)
```

[ ]:
```

```