main

August 24, 2021

1 Test for enrichment of mashr results with DTU

[1]:
```
import functools
import numpy as np
import pandas as pd
from pybiomart import Dataset
from scipy.stats import fisher_exact
from statsmodels.stats.multitest import multipletests
```

1.1 Function

1.1.1 Cached functions

[2]:
```
@functools.lru_cache()
def get_mashr_results(feature):
    return pd.read_csv("../../_m/%s/brainseq_ancestry_4tissues_mashr.tsv" %
                       feature, sep='\t')


@functools.lru_cache()
def get_mashr_results_by_tissue(tissue, feature):
    return get_mashr_results(feature).loc[:, ["Feature", tissue]]


@functools.lru_cache()
def get_mashr_sig_genes(tissue, feature):
    return get_mashr_results(feature)[(get_mashr_results(feature)[tissue] ==␣
 ↪1)].loc[:, ["Feature", tissue]]


@functools.lru_cache()
def get_mashr_tissue_specific(tissue, feature):
    return␣
 ↪get_mashr_results(feature)[(get_mashr_results(feature)["N_Regions_Shared"]␣
 ↪== 1) &
                                       (get_mashr_results(feature)[tissue] ==␣
 ↪1)].loc[:, ["Feature", tissue]]
```

```
@functools.lru_cache()
def feature_map(feature):
    return {"genes": "Gene", "transcripts": "Transcript",
            "exons": "Exon", "junctions": "Junction"}[feature]


@functools.lru_cache()
def tissue_map(tissue):
    return {"Caudate": "caudate", "Dentate Gyrus": "dentateGyrus",
            "DLPFC": "dlpfc", "Hippocampus": "hippocampus"}[tissue]


@functools.lru_cache()
def get_annot(tissue, feature):
    fn = "../../../../%s/_m/%s/diffExpr_EAvsAA_full.txt" % (tissue_map(tissue),␣
 ↪feature)
    symbols = {"gene_name": "Symbol", "newGeneSymbol": "Symbol", "index":␣
 ↪"Feature"}
    if tissue == "Caudate":
        tx = "gene_name"
    else:
        tx = "Symbol"
    symbol_map = {"transcripts": tx, "junctions": "newGeneSymbol",
                  "genes": "Symbol", "exons": "Symbol"}
    return pd.read_csv(fn, sep='\t', index_col=0).loc[:, [symbol_map[feature]]]\
             .reset_index().rename(columns=symbols)


@functools.lru_cache()
def get_DS(tissue):
    ## Load local splicing results
    return pd.read_csv("../../../../%s/localsplicing/visualization/_m/
 ↪cluster_ds_results_annotated.txt" %
                       tissue_map(tissue), sep='\t')\
             .rename(columns={"gene": "Symbol"})


@functools.lru_cache()
def annotate_ds(tissue, feature):
    df = get_DS(tissue).merge(get_annot(tissue, feature), on="Symbol")
    df[tissue] = 1
    return df
```

1.1.2 Simple functions

[3]:
```
def print_overlap(feature, tissue, fnc):
    overlap = len(set(fnc(tissue, feature).Feature) &
                  set(annotate_ds(tissue, feature).Feature))
    print("There are {} ({:.1%}) DS overlapping DE {}."\
          .format(overlap, overlap/len(annotate_ds(tissue, feature).Feature.
 ↪unique()), feature))


def merge_data(feature, tissue):
    return annotate_ds(tissue, feature).loc[:, ["Feature", tissue,␣
 ↪"annotation"]]\
                                       .drop_duplicates(subset="Feature")\
                                       .
 ↪merge(get_mashr_results_by_tissue(tissue, feature),
                                              on="Feature", how="outer",␣
 ↪suffixes=["_ds", "_mashr"])\
                                       .fillna(0)


def cal_fishers(feature, tissue):
    dt = merge_data(feature, tissue)
    table = [[np.sum((dt['%s_ds' % tissue]==1) & ((dt['%s_mashr' %␣
 ↪tissue]==1))),
              np.sum((dt['%s_ds' % tissue]==1) & ((dt['%s_mashr' %␣
 ↪tissue]==0)))],
             [np.sum((dt['%s_ds' % tissue]==0) & ((dt['%s_mashr' %␣
 ↪tissue]==1))),
              np.sum((dt['%s_ds' % tissue]==0) & ((dt['%s_mashr' %␣
 ↪tissue]==0)))]]
    #print(table)
    return fisher_exact(table)
```

1.2 Overlap with DE

[4]:
```
for tissue in ["Caudate", "Dentate Gyrus", "DLPFC", "Hippocampus"]:
    print("{}".format(tissue))
    for feature in ["genes", "transcripts", "exons", "junctions"]:
        print_overlap(feature, tissue, get_mashr_sig_genes)
    print("\n{}-specific".format(tissue))
    for feature in ["genes", "transcripts", "exons", "junctions"]:
        print_overlap(feature, tissue, get_mashr_tissue_specific)
    print("")
```

```
Caudate
There are 398 (27.7%) DS overlapping DE genes.
```

```
There are 1552 (14.4%) DS overlapping DE transcripts.
There are 7702 (17.1%) DS overlapping DE exons.
There are 2918 (13.5%) DS overlapping DE junctions.

Caudate-specific
There are 71 (4.9%) DS overlapping DE genes.
There are 271 (2.5%) DS overlapping DE transcripts.
There are 1551 (3.4%) DS overlapping DE exons.
There are 547 (2.5%) DS overlapping DE junctions.

Dentate Gyrus
There are 111 (22.4%) DS overlapping DE genes.
There are 650 (21.2%) DS overlapping DE transcripts.
There are 2629 (15.7%) DS overlapping DE exons.
There are 1326 (15.9%) DS overlapping DE junctions.

Dentate Gyrus-specific
There are 11 (2.2%) DS overlapping DE genes.
There are 132 (4.3%) DS overlapping DE transcripts.
There are 546 (3.3%) DS overlapping DE exons.
There are 285 (3.4%) DS overlapping DE junctions.

DLPFC
There are 300 (28.7%) DS overlapping DE genes.
There are 1338 (18.9%) DS overlapping DE transcripts.
There are 6291 (19.1%) DS overlapping DE exons.
There are 2611 (16.4%) DS overlapping DE junctions.

DLPFC-specific
There are 41 (3.9%) DS overlapping DE genes.
There are 214 (3.0%) DS overlapping DE transcripts.
There are 1098 (3.3%) DS overlapping DE exons.
There are 440 (2.8%) DS overlapping DE junctions.

Hippocampus
There are 279 (26.3%) DS overlapping DE genes.
There are 1443 (20.4%) DS overlapping DE transcripts.
There are 5979 (17.8%) DS overlapping DE exons.
There are 2406 (15.5%) DS overlapping DE junctions.

Hippocampus-specific
There are 50 (4.7%) DS overlapping DE genes.
There are 280 (4.0%) DS overlapping DE transcripts.
There are 1155 (3.4%) DS overlapping DE exons.
There are 444 (2.9%) DS overlapping DE junctions.
```

1.3 Enrichment analysis

[5]:
```
feature_lt = []; pval_lt = []; oddratio_lt = []; tissue_lt = []
for tissue in ["Caudate", "Dentate Gyrus", "DLPFC", "Hippocampus"]:
    print(tissue)
    for feature in ["genes", "transcripts", "exons", "junctions"]:
        oddratio, pval = cal_fishers(feature, tissue)
        feature_lt.append(feature_map(feature))
        pval_lt.append(pval)
        oddratio_lt.append(oddratio)
        tissue_lt.append(tissue)
        print("Enrichment of DS within DE for {}:\n\tOdd Ratio: {:.2f}; P-value:
 ↪ {:.1e}"\
              .format(feature_map(feature), oddratio, pval))
    print("")
pval_lt = [1e-323 if x == 0 else x for x in pval_lt]
_, fdr, _, _ = multipletests(pval_lt, method='fdr_bh')
df = pd.DataFrame({"Tissue": tissue_lt, "Feature": feature_lt,
                   "OR": oddratio_lt, "PValue": pval_lt, "FDR": fdr})
```

```
Caudate
Enrichment of DS within DE for Gene:
        Odd Ratio: 1.09; P-value: 1.5e-01
Enrichment of DS within DE for Transcript:
        Odd Ratio: 0.93; P-value: 1.1e-02
Enrichment of DS within DE for Exon:
        Odd Ratio: 1.45; P-value: 1.9e-146
Enrichment of DS within DE for Junction:
        Odd Ratio: 1.52; P-value: 1.8e-70

Dentate Gyrus
Enrichment of DS within DE for Gene:
        Odd Ratio: 1.03; P-value: 7.8e-01
Enrichment of DS within DE for Transcript:
        Odd Ratio: 1.15; P-value: 2.5e-03
Enrichment of DS within DE for Exon:
        Odd Ratio: 1.55; P-value: 9.7e-80
Enrichment of DS within DE for Junction:
        Odd Ratio: 1.97; P-value: 4.2e-89

DLPFC
Enrichment of DS within DE for Gene:
        Odd Ratio: 1.16; P-value: 4.2e-02
Enrichment of DS within DE for Transcript:
        Odd Ratio: 1.07; P-value: 3.7e-02
Enrichment of DS within DE for Exon:
        Odd Ratio: 1.56; P-value: 3.4e-176
Enrichment of DS within DE for Junction:
```

```
        Odd Ratio: 1.78; P-value: 7.3e-119

Hippocampus
Enrichment of DS within DE for Gene:
        Odd Ratio: 1.00; P-value: 1.0e+00
Enrichment of DS within DE for Transcript:
        Odd Ratio: 1.11; P-value: 8.8e-04
Enrichment of DS within DE for Exon:
        Odd Ratio: 1.47; P-value: 5.1e-128
Enrichment of DS within DE for Junction:
        Odd Ratio: 1.72; P-value: 6.0e-100
```

[6]:
```
df.to_csv('diffSplice_enrichment_analysis.txt', sep='\t', index=False)
```

[ ]:
```

```