main

July 12, 2021

1 Enrichment in DE genes

[1]:

```
import functools
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import fisher_exact
from statsmodels.stats.multitest import multipletests
```

1.1 Functions

1.1.1 Cached functions

[2]:

```
@functools.lru_cache()
def get_wgcna_modules():
    return pd.read_csv("../../_m/modules.csv", index_col=0)


@functools.lru_cache()
def get_degs():
    return set(pd.read_csv('../../../../differential_analysis/'+\
                           'dentateGyrus/_m/genes/diffExpr_EAvsAA_FDR05.txt',
                           sep='\t', usecols=[0], index_col=0).index)


@functools.lru_cache()
def get_mhc_genes():
    return set(pd.read_csv('../../../../input/counts/mhc_region_genes/'+\
                           '_m/mhc_genes.csv')['gene_id'])
```

1.1.2 Simple functions

[3]:

```
def fet(a, b, u):
    # a, b, u are sets
    # u is the universe
    yes_a = u.intersection(a)
    yes_b = u.intersection(b)
    no_a = u - a
    no_b = u - b
    m = [[len(yes_a.intersection(yes_b)), len(no_a.intersection(yes_b)) ],
         [len(yes_a.intersection(no_b)), len(no_a.intersection(no_b))]]
    return fisher_exact(m)


def enrichment_rows():
    mod = get_wgcna_modules().module.unique()
    u = set(get_wgcna_modules().index)
    for ii in range(len(mod)): # for each module
        a = set(get_wgcna_modules()[(get_wgcna_modules().module) == mod[ii]].
 ↪index)
        b = set(get_wgcna_modules()[(get_wgcna_modules().module) == mod[ii]].
 ↪index) - get_mhc_genes()
        yield (mod[ii],
               len(a),
               *fet(a, get_degs(), u),
               *fet(b, get_degs() - get_mhc_genes(), u),
               )
```

1.2 Main

1.2.1 Enrichment

[4]:

```
edf = pd.DataFrame.from_records(enrichment_rows(),
                                 columns=['Module_ID', 'N_Genes', 'DEG_OR',␣
 ↪'DEG_P',
                                          'DEG_noMHC_OR', 'DEG_noMHC_P'],
                                 index='Module_ID')
edf['DEG_FDR'] = multipletests(edf['DEG_P'], method='fdr_bh')[1]
edf['DEG_noMHC_FDR'] = multipletests(edf['DEG_noMHC_P'], method='fdr_bh')[1]
edf = edf.loc[:, ['N_Genes', 'DEG_OR', 'DEG_P', 'DEG_FDR', 'DEG_noMHC_OR',␣
 ↪'DEG_noMHC_P', 'DEG_noMHC_FDR']]
```

[5]:

```
print(edf[(edf["DEG_FDR"] < 0.05)].shape)
edf[(edf["DEG_FDR"] < 0.05)]
```

```
(20, 7)
```

[5]:

```
                N_Genes    DEG_OR         DEG_P       DEG_FDR  DEG_noMHC_OR  \
Module_ID
greenyellow         445  0.289751  1.381787e-03  4.204116e-03      0.295425
brown              1583  0.583540  1.120013e-03  3.970956e-03      0.603106
grey               6097  1.281627  1.509170e-03  4.204116e-03      1.293676
darkgreen           127  0.000000  1.584131e-02  3.251638e-02      0.000000
lightcyan           227  2.830277  4.221828e-05  2.744188e-04      2.808123
pink                489  0.316468  1.474444e-03  4.204116e-03      0.321855
```

```
magenta             473   2.051053  2.895736e-04  1.411671e-03      2.082650
grey60              190  19.004695  2.517402e-58  9.817866e-57     19.324708
salmon              363   0.070352  3.077483e-05  2.400437e-04      0.071593
yellow             1255   0.467871  1.084722e-04  6.043449e-04      0.477877
darkgrey            104   4.429896  6.889969e-06  1.080747e-04      4.547772
black               490   1.843652  2.246851e-03  5.476700e-03      1.774039
green              1061   0.359122  8.313437e-06  1.080747e-04      0.364895
red                 906   1.617648  2.132958e-03  5.476700e-03      1.643186
purple              465   0.447695  1.784913e-02  3.480581e-02      0.454366
skyblue              90   4.428706  2.749779e-05  2.400437e-04      4.495074
lightyellow         156   0.000000  4.744847e-03  1.088524e-02      0.000000
midnightblue        260   0.198703  7.178820e-03  1.555411e-02      0.204043
darkolivegreen       56   4.350043  1.031264e-03  3.970956e-03      4.508974
steelblue            84   3.531624  1.097446e-03  3.970956e-03      3.584349

                 DEG_noMHC_P  DEG_noMHC_FDR
Module_ID
greenyellow     1.910709e-03   5.678946e-03
brown           2.504022e-03   6.510456e-03
grey            1.030448e-03   3.653406e-03
darkgreen       1.559177e-02   3.200415e-02
lightcyan       6.819974e-05   4.432983e-04
pink            2.038596e-03   5.678946e-03
magenta         2.610261e-04   1.272502e-03
grey60          1.395232e-56   5.441405e-55
salmon          2.971984e-05   2.318147e-04
yellow          1.772028e-04   9.872728e-04
darkgrey        5.151680e-06   1.004578e-04
black           4.567819e-03   1.113406e-02
green           1.088016e-05   1.414421e-04
red             1.462311e-03   4.752511e-03
purple          2.353411e-02   4.589152e-02
skyblue         2.374607e-05   2.315242e-04
lightyellow     7.573192e-03   1.640858e-02
midnightblue    6.902130e-03   1.583430e-02
darkolivegreen  8.335515e-04   3.612056e-03
steelblue       9.866607e-04   3.653406e-03
```

[6]:

```
print(edf[(edf["DEG_noMHC_FDR"] < 0.05)].shape)
set(edf[(edf["DEG_FDR"] < 0.05)].index) - set(edf[(edf["DEG_noMHC_FDR"] < 0.
 ↪05)].index)
```

```
(20, 7)
```

[6]:

```
set()
```

sienna3 is enriched in MHC differentially expressed genes

[7]:

```
edf.to_csv('wgcna_module_enrichment.csv')
```

1.2.2 Plot heatmap

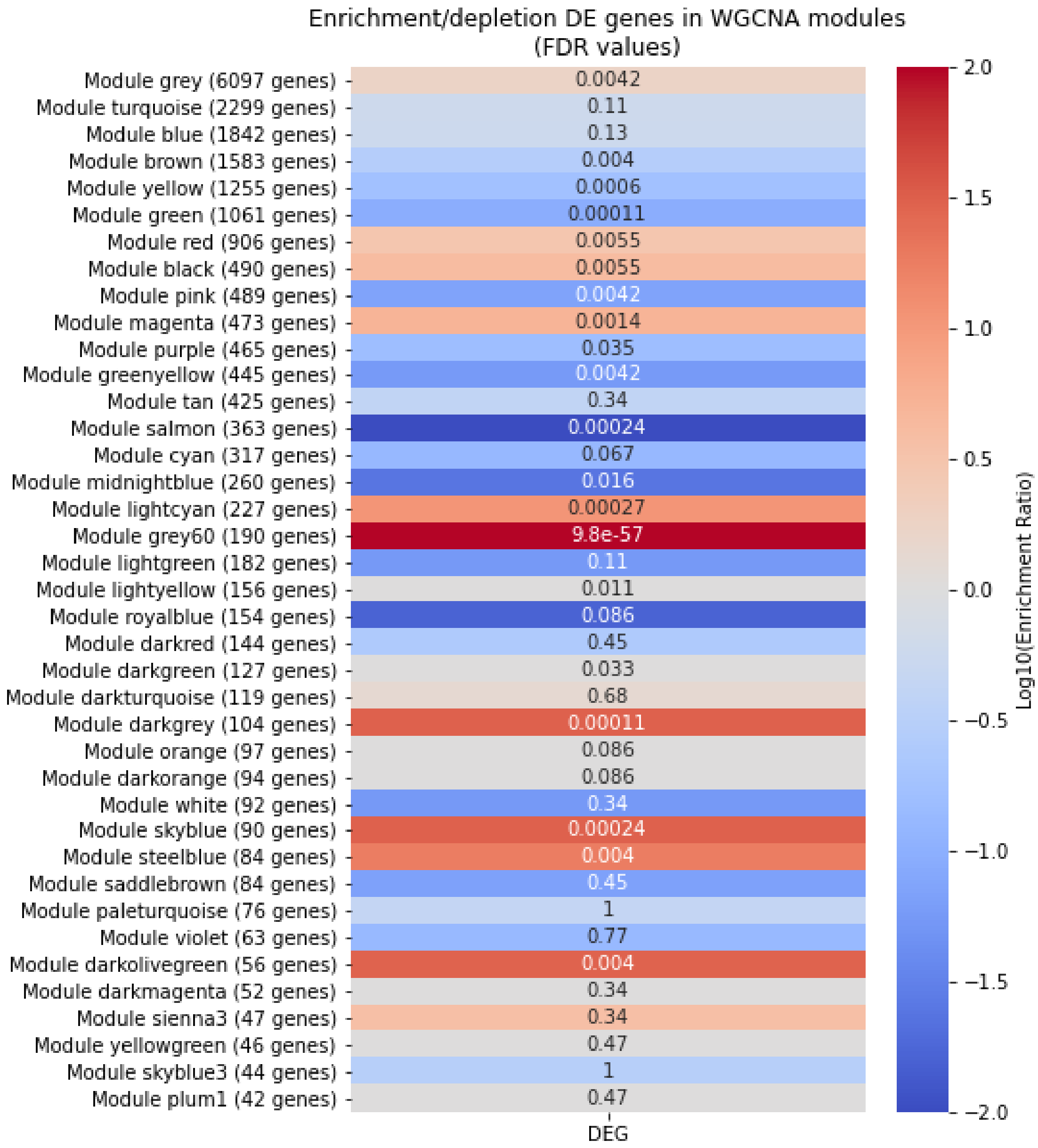
[8]:

```
df = edf.sort_values("N_Genes", ascending=False)
df2 = np.log(df.loc[:, ['DEG_OR']]).replace([np.inf, -np.inf], 0)
df2.columns = ['DEG']
df2.index = ["Module %s (%d genes)" % (x,y) for x,y in zip(df2.index,␣
 ↪df['N_Genes'])]
df3 = df.loc[:, ['DEG_FDR']]

fig, ax = plt.subplots(figsize=(6,10))
p = sns.heatmap(df2, cmap='coolwarm', annot=df3, yticklabels=df2.index,␣
 ↪center=0,
                cbar_kws={'label': 'Log10(Enrichment Ratio)'}, vmin=-2, vmax=2)
p.set_title("Enrichment/depletion DE genes in WGCNA modules\n(FDR values)")
p.get_figure().savefig('wgcna_module_enrichment.pdf', bbox_inches='tight')
p
```

[8]:

```
<AxesSubplot:title={'center':'Enrichment/depletion DE genes in WGCNA
modules\n(FDR values)'}>
```



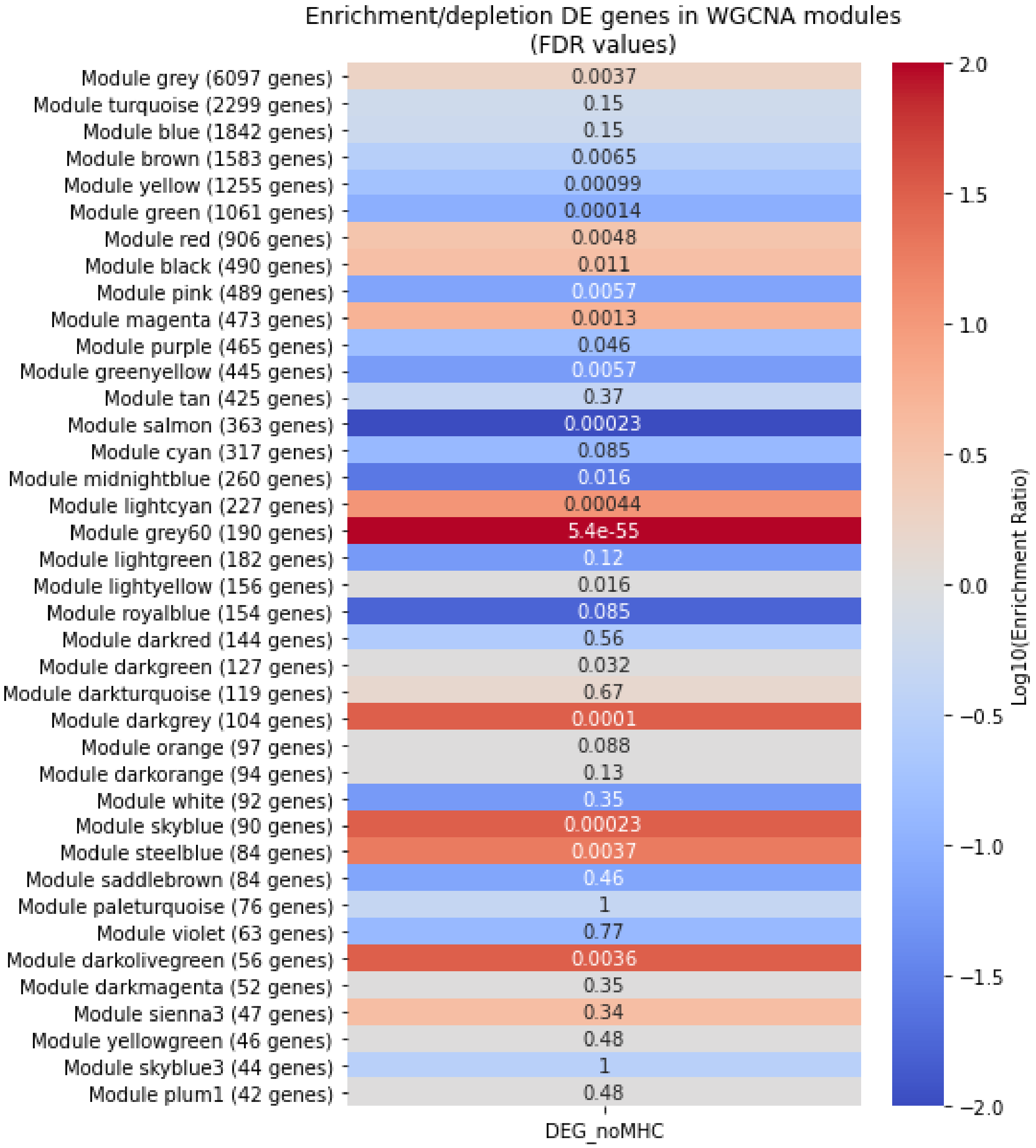
[9]:

```
df = edf.sort_values("N_Genes", ascending=False)
df2 = np.log(df.loc[:, ['DEG_noMHC_OR']]).replace([np.inf, -np.inf], 0)
df2.columns = ['DEG_noMHC']
df2.index = ["Module %s (%d genes)" % (x,y) for x,y in zip(df2.index,␣
 ↪df['N_Genes'])]
df3 = df.loc[:, ['DEG_noMHC_FDR']]

fig, ax = plt.subplots(figsize=(6,10))
```

```
p = sns.heatmap(df2, cmap='coolwarm', annot=df3, yticklabels=df2.index,␣
 ↪center=0,
                 cbar_kws={'label': 'Log10(Enrichment Ratio)'}, vmin=-2, vmax=2)
p.set_title("Enrichment/depletion DE genes in WGCNA modules\n(FDR values)")
p.get_figure().savefig('wgcna_module_enrichment_noMHC.pdf', bbox_inches='tight')
p
```

```
[9]: <AxesSubplot:title={'center':'Enrichment/depletion DE genes in WGCNA
     modules\n(FDR values)'}>
```



[ ]:

```
```