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/'+\
                           'caudate/_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)]
```

```
(14, 7)
```

[5]:
```
                N_Genes    DEG_OR         DEG_P       DEG_FDR  DEG_noMHC_OR  \
Module_ID
grey              12944  1.440224  3.448904e-19  1.276095e-17      1.464040
blue               1192  0.720031  6.122328e-04  3.236088e-03      0.718478
violet               56  2.397301  8.508951e-03  2.862102e-02      2.615486
turquoise          1761  0.727306  5.599962e-05  6.906620e-04      0.739907
white                83  0.244237  5.375600e-03  2.209969e-02      0.166649
yellow              447  1.667279  4.125988e-05  6.906620e-04      1.686123
```

```
skyblue              77  1.999277  1.671890e-02  4.458767e-02      2.021343
darkmagenta          52  0.000000  1.421483e-03  6.574357e-03      0.000000
brown              1030  0.700099  5.874883e-04  3.236088e-03      0.709675
sienna3              47  0.141740  1.687101e-02  4.458767e-02      0.153318
lightgreen          160  0.297254  2.563811e-04  1.897220e-03      0.304539
black               316  0.437655  1.080762e-04  9.997052e-04      0.442496
darkolivegreen       55  0.120692  8.070029e-03  2.862102e-02      0.122017
midnightblue        213  0.528084  1.087028e-02  3.351671e-02      0.533914

                 DEG_noMHC_P  DEG_noMHC_FDR
Module_ID
grey            9.921223e-21   3.670853e-19
blue            6.614695e-04   4.079062e-03
violet          3.264516e-03   1.207871e-02
turquoise       1.472774e-04   1.362316e-03
white           1.533300e-03   6.303567e-03
yellow          3.678111e-05   6.804505e-04
skyblue         1.624403e-02   4.623300e-02
darkmagenta     1.328575e-03   6.144659e-03
brown           9.313964e-04   4.923095e-03
sienna3         2.495277e-02   6.594661e-02
lightgreen      5.137976e-04   3.802103e-03
black           1.068366e-04   1.317652e-03
darkolivegreen  7.959279e-03   2.677212e-02
midnightblue    1.396602e-02   4.306189e-02
```

[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)
```

```
(13, 7)
```

[6]:
```
{'sienna3'}
```

sienna3 is enriched in MHC differentially expressed genes

[7]:
```
edf.to_csv('wgcna_module_enrichment.csv')
```

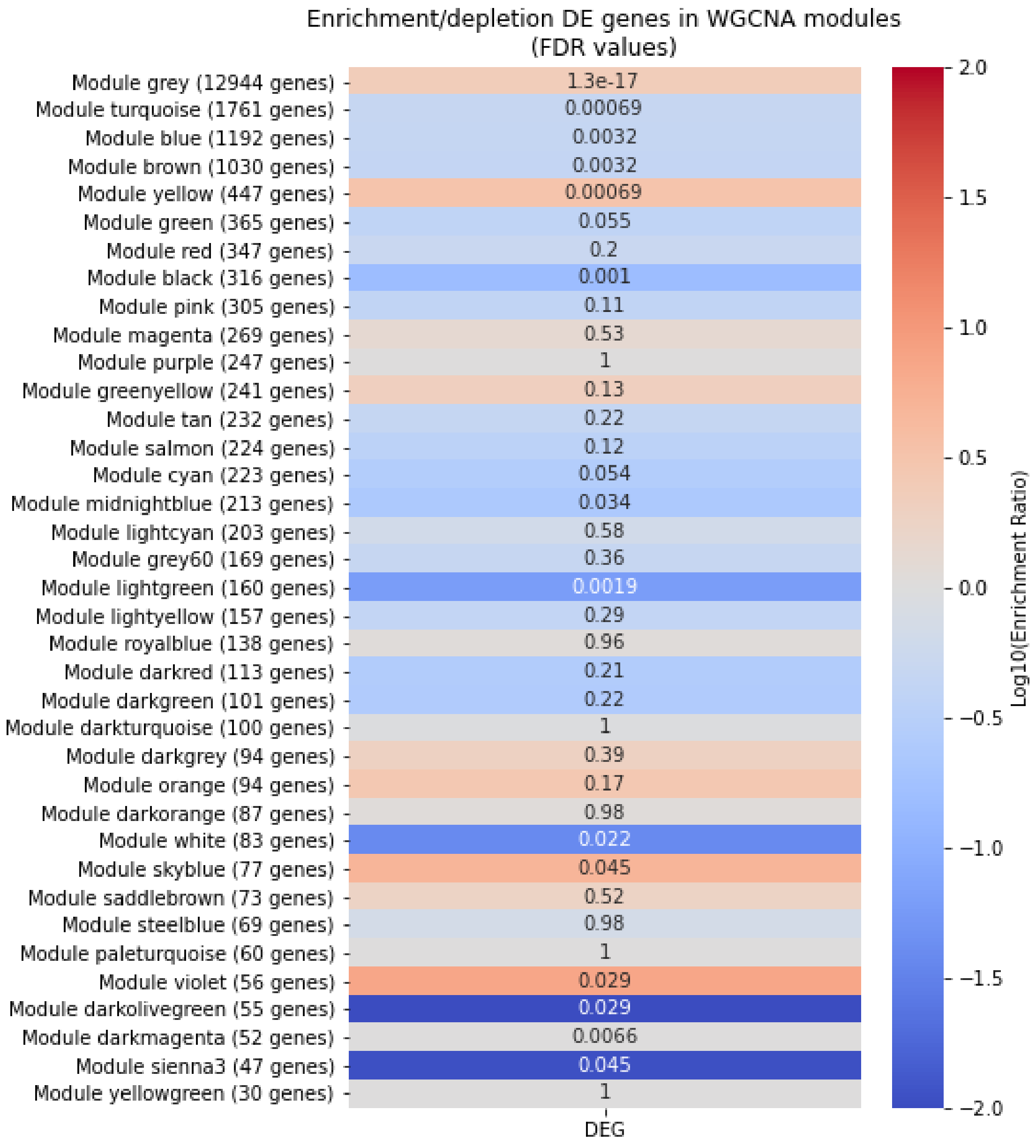
1.2.2 Plot heatmap

[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
```

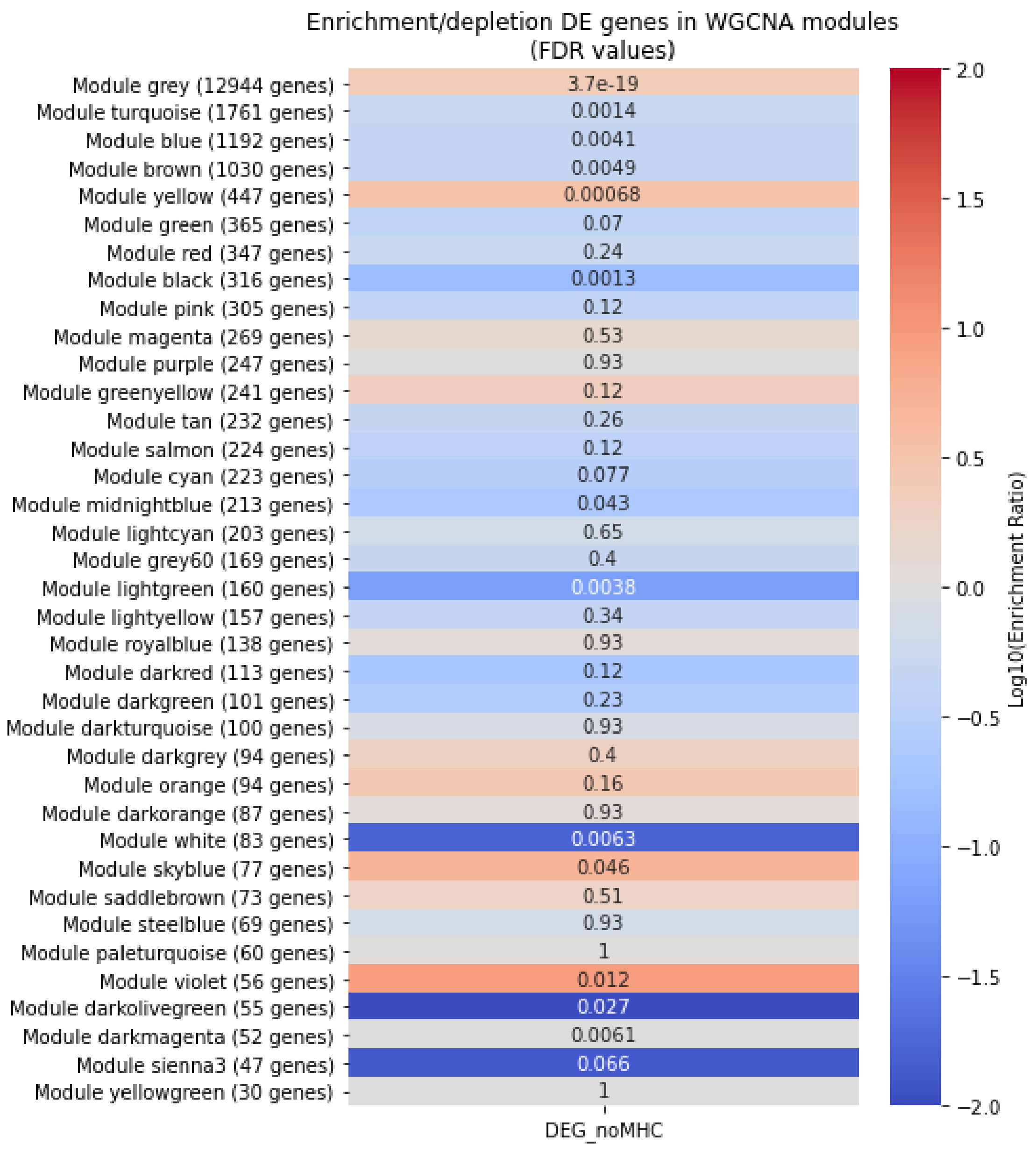
[8]:
```
<AxesSubplot:title={'center':'Enrichment/depletion DE genes in WGCNA
modules\n(FDR values)'}>
```



[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)'}>
```


Enrichment/depletion DE genes in WGCNA modules
(FDR values)
Module grey (12944 genes)
3.7e-19
Module turquoise (1761 genes)
0.0014
Module blue (1192 genes)
0.0041
Module brown (1030 genes)
0.0049
Module yellow (447 genes)
0.00068
Module green (365 genes)
0.07
Module red (347 genes)
0.24
Module black (316 genes)
0.0013
Module pink (305 genes)
0.12
Module magenta (269 genes)
0.53
Module purple (247 genes)
0.93
Module greenyellow (241 genes)
0.12
Module tan (232 genes)
0.26
Module salmon (224 genes)
0.12
Module cyan (223 genes)
0.077
Module midnightblue (213 genes)
0.043
Module lightcyan (203 genes)
0.65
Module grey60 (169 genes)
0.4
Module lightgreen (160 genes)
0.0038
Module lightyellow (157 genes)
0.34
Module royalblue (138 genes)
0.93
Module darkred (113 genes)
0.12
Module darkgreen (101 genes)
0.23
Module darkturquoise (100 genes)
0.93
Module darkgrey (94 genes)
0.4
Module orange (94 genes)
0.16
Module darkorange (87 genes)
0.93
Module white (83 genes)
0.0063
Module skyblue (77 genes)
0.046
Module saddlebrown (73 genes)
0.51
Module steelblue (69 genes)
0.93
Module paleturquoise (60 genes)
1
Module violet (56 genes)
0.012
Module darkolivegreen (55 genes)
0.027
Module darkmagenta (52 genes)
0.0061
Module sienna3 (47 genes)
0.066
Module yellowgreen (30 genes)
1
DEG_noMHC
Log10(Enrichment Ratio)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0

```
[ ]:
```