04.0_TissueEnrichment

March 23, 2022

1 Tissue-specificity of the ascr#10 response

In this notebook, we explore the tissue specificity of the ascr#10 response in *C. elegans*. Please note that most enrichment analyses first count the number of DE genes associated with a specific term, and compare that number with the expected number under a uniform random model–this is called a hypergeometric test. In this notebook, we do NOT perform hypergeometric tests to identify enriched tissues. We take a different approach.

In this notebook, we download the gene expression patterns from Wormbase, and then we test whether genes that are expressed in a given tissue are going UP(down) more often than expected by random chance. We do this using a binomial test.

The WormBase gene expression-tissue database has been processed in a fairly specific way. First, we removed all non-DE genes at 50 or at 58hrs from the database. Next, we dropped all tissues that did not have at least 5 genes annotated to them. Finally, we removed 'promiscuous genes', genes that are annotated to many (>30) tissues. Originally, the database had 1,535 tissues and after processing we have 264 tissues.

[1]:
```
import sys
sys.path.append('../python')
import tissue_utils as utils  # the functions I wrote are here
import json
import pandas as pd
import numpy as np
import scipy
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

from statsmodels.stats.multitest import fdrcorrection

from matplotlib import rc
rc('text', usetex=False)
# rc('text.latex', preamble=r'\usepackage{cmbright}')
rc('font', **{'family': 'sans-serif', 'sans-serif': ['Helvetica']})

%matplotlib inline

# This enables SVG graphics inline.
```

```
%config InlineBackend.figure_formats = {'png', 'retina'}

rc = {'lines.linewidth': 2,
      'axes.labelsize': 25,
      'axes.titlesize': 25,
      'axes.facecolor': 'DFDFE5'}
sns.set_context('notebook', rc=rc)
sns.set_style("dark")

mpl.rcParams['xtick.labelsize'] = 18
mpl.rcParams['ytick.labelsize'] = 18
mpl.rcParams['legend.fontsize'] = 20
# load stuff:
res = pd.read_csv('../data/master_table.tsv', sep='\t', index_col=0)
# keep DE genes only:
res = res[((res['padj-58'] < 0.05) | (res['padj-50'] < 0.05)) & (res['Sign-WT']␣
 ↪== 'Same')]

cols = ['externalgenename', 'log2FoldChange{0}', 'padj{0}']
fetch = lambda x: [c.format(x) for c in cols]
rename = lambda x: {'log2FoldChange{0}'.format(x): 'log2FoldChange', 'padj{0}'.
 ↪format(x): 'padj'}

# data wrangling to create a new dataframe that contains the `results` data,␣
 ↪but will also include
# the tissue annotations
a = res[fetch('-58')].rename(columns=rename('-58'))
b = res[fetch('-50')].rename(columns=rename('-50'))
a['data'] = '58hrs'
b['data'] = '50hrs'
annotated_data = pd.concat([a[a.padj < 0.05], b[b.padj < 0.05]])
cat_type = pd.CategoricalDtype(categories=['50hrs', '58hrs'], ordered=True)
annotated_data.data = annotated_data.data.astype(cat_type)

# load tissue dictionary:
tissues = utils.load_tissues(res.index, 5, 30)
annotated_data = annotated_data.join(tissues.set_index('wbid')).
 ↪dropna(subset=['tissue'])
annotated_data.head()
# res
```

```
tissues originally: 1535
tissues afterwards: 264
```

```
[1]:                externalgenename  log2FoldChange      padj   data  \
     WBGene00000001            aap-1        0.096256  0.009407  58hrs
     WBGene00000001            aap-1        0.096256  0.009407  58hrs
```

```
WBGene00000001            aap-1        0.096256  0.009407  58hrs
WBGene00000001            aap-1        0.096256  0.009407  58hrs
WBGene00000023            abt-5       -0.342206  0.000949  50hrs

               external_gene_name   target_id                 species  \
WBGene00000001              aap-1  Y110A7A.10  Caenorhabditis elegans
WBGene00000001              aap-1  Y110A7A.10  Caenorhabditis elegans
WBGene00000001              aap-1  Y110A7A.10  Caenorhabditis elegans
WBGene00000001              aap-1  Y110A7A.10  Caenorhabditis elegans
WBGene00000023              abt-5   Y53C10A.9  Caenorhabditis elegans

                               tissue
WBGene00000001  body wall musculature
WBGene00000001             hypodermis
WBGene00000001              intestine
WBGene00000001                 neuron
WBGene00000023                   head
```

1.1 Testing the tissues.

In the cell above, we loaded our data, and we also loaded our gene expression pattern database into a variable called `tissues`. Here, we perform a binomial test on each of the tissue programs.

[2]:

```
# perform a binomial test for change in direction:
alpha = 0.05
data = utils.test_tissue_direction(res[res['padj-58'] < 0.05], tissues)
data_50 = utils.test_tissue_direction(res[res['padj-50'] < 0.05], tissues,␣
 ↪col='log2FoldChange-50')

data = pd.concat([data, data_50], keys=['58hrs', '50hrs']
                ).reset_index().rename(
                columns={'level_0': 'data'}
                ).drop('level_1', axis=1)

data = utils.fdr_correct(data)
data.sort_values('fdr', inplace=True)

data['MeanFracPos'] = data.groupby(['tissue', 'data']).FracPos.transform(np.
 ↪mean)
data['MeanFracPos'] = data.groupby(['tissue', 'data']).FracPos.transform(np.
 ↪mean)

# keep only significant results:
keep = data.groupby('tissue').sig.sum()
data = data[data.tissue.isin(keep[keep > 0].index)]
```

```
m = 'There were {0} tissues that changed more in one direction than expected by␣
 ↪random chance in dataset {1}'
for n, g in data.groupby('data'):
    print(m.format(len(g), n))

data
```

```
There were 46 tissues that changed more in one direction than expected by random
chance in dataset 50hrs
There were 49 tissues that changed more in one direction than expected by random
chance in dataset 58hrs
```

```
[2]:      data                 tissue          pval   FracPos  FracPosExpected  \
     0    58hrs              germ line  2.449246e-24  0.953125         0.646552
     1    58hrs                   Cell  1.625358e-11  0.938776         0.646552
     2    58hrs         nervous system  2.933895e-10  0.828000         0.646552
     3    58hrs  body wall musculature  8.520268e-10  0.808581         0.646552
     194  50hrs    reproductive system  1.626558e-09  0.901099         0.615059
     ..     …                      …           …        …               …
     102  58hrs              spermatid  2.527991e-01  0.428571         0.646552
     269  50hrs                   hyp9  6.819123e-01  0.500000         0.615059
     263  50hrs                  hyp11  6.819123e-01  0.500000         0.615059
     270  50hrs            somatic cell  7.188394e-01  0.750000         0.615059
     286  50hrs         phasmid neuron  1.000000e+00  0.600000         0.615059

                   fdr    neglogq    sig  MeanFracPos
     0    7.151798e-22  21.145585   True     0.953125
     1    2.373022e-09   8.624698   True     0.938776
     2    2.855658e-08   7.544294   True     0.828000
     3    6.219796e-08   7.206224   True     0.808581
     194  9.499100e-08   7.022318   True     0.901099
     ..            …         …     …          …
     102  4.824663e-01   0.316533  False     0.428571
     269  8.473123e-01   0.071956  False     0.500000
     263  8.473123e-01   0.071956  False     0.500000
     270  8.856587e-01   0.052734  False     0.750000
     286  1.000000e+00  -0.000000  False     0.600000

     [95 rows x 9 columns]
```

Next, let's remove uninformative terms like `cell`, `tail` or `head`. I will also remove highly similar terms to prevent too much redundancy. For completeness, I've made sure to print all tissues that I will remove:

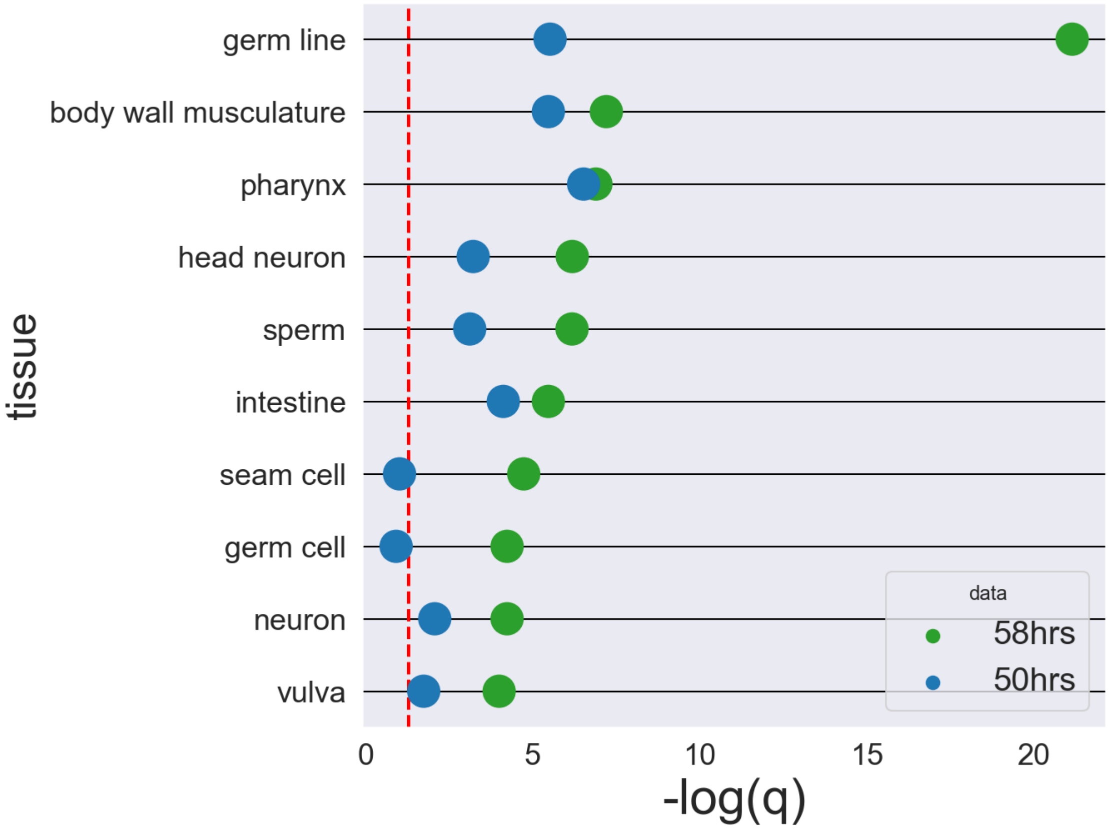
```
[3]: remove = ['Cell', 'tail', 'head', 'male gonad', 'gonad', ]  # so broad. why␣
      ↪ever have these terms...
     remove = utils.similarity_trimming(data.tissue.unique(), tissues, remove)
```

```
print('The following list of tissues will be removed from the plot',␣
 ↪list(set(remove)))
```

```
The following list of tissues will be removed from the plot ['Psub2', 'Cell',
'EMS', 'male gonad', 'ABp', 'hypodermis', 'gonad', 'AB', 'reproductive system',
'nervous system', 'posterior distal tip cell', 'anterior distal tip cell', 'mu-
int-L', 'body region', 'vulval muscle', 'Psub1', 'Z2', 'hyp12', 'Z3', 'ABa',
'rectal epithelium', 'head', 'P0', 'hermaphrodite gonad', 'anal depressor
muscle', 'phasmid neuron', 'hyp11', 'ventral cord neuron', 'excretory cell',
'mu-int-R', 'Psub3', 'tail', 'tail neuron']
```

Let's plot the results:

[9]:
```
to_plot = data[(~data.tissue.isin(remove)) & (data.sig == True)].head(15)#.
 ↪sort_values('fdr').head(10)
to_plot = data[(data.tissue.isin(to_plot.tissue))]

tissues_plotted = to_plot.tissue.unique()
fig, ax = utils.pretty_GSEA_plots(to_plot, annotated_data, alpha, size=20)
# ax.legend(loc=(0.1, .7))
ax.yaxis.grid(color='black', linewidth=1)
ax.set_xlabel('-log(q)', fontsize=30)
# ax.set_xlabel('Fraction of Genes Upregulated', fontsize=30)
plt.savefig('../figs/tissue_GSEA.svg', bbox_inches='tight', transparent=False)
```



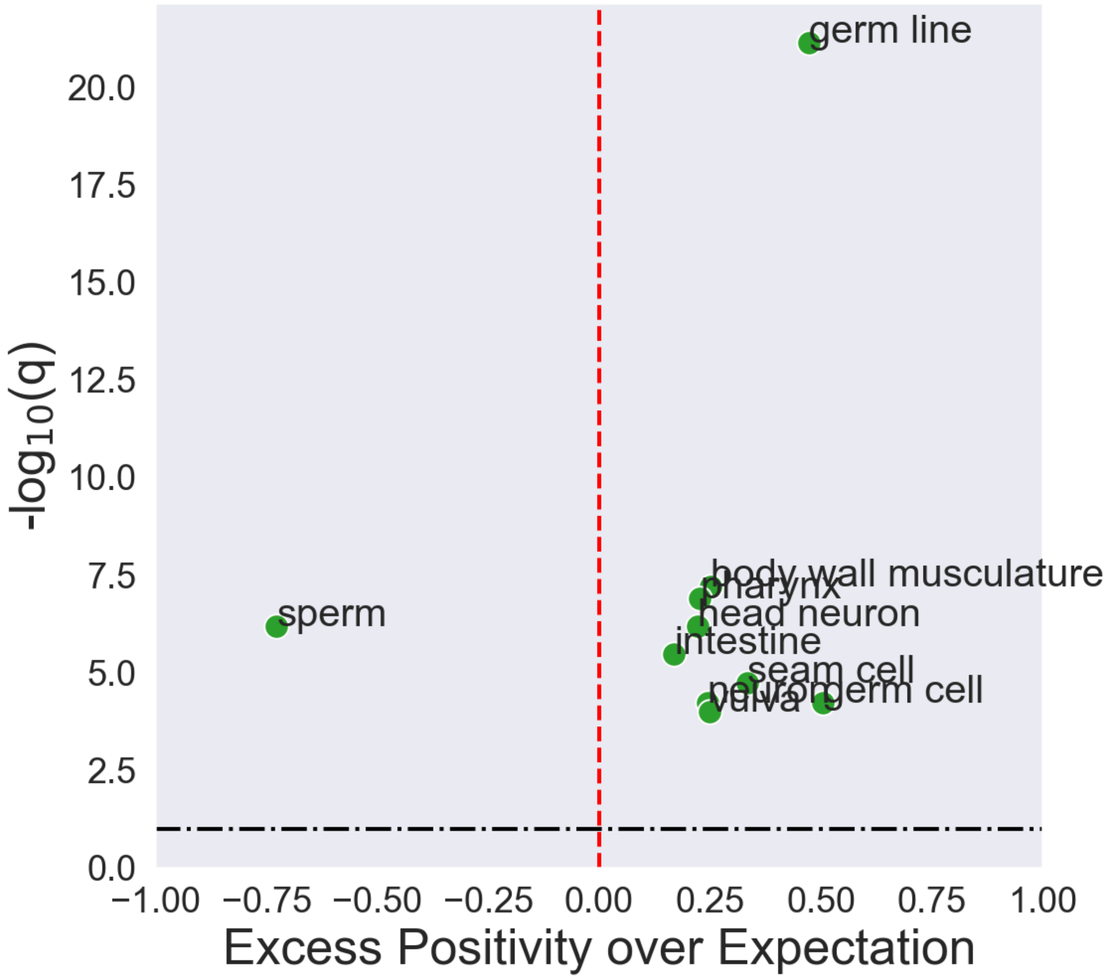
1.2 Volcano Plot

Here is the volcano plot that was reproduced in the paper:

[15]:

```python
to_plot['Excess'] = (to_plot.FracPos - to_plot.FracPosExpected) / to_plot.FracPosExpected

fig, ax = plt.subplots(figsize=(8, 8))
sns.scatterplot(x='Excess', y='neglogq', data=to_plot[to_plot.data == '58hrs'], s=150, color='tab:green', ax=ax)

for tissue, df in to_plot[to_plot.data == '58hrs'].groupby('tissue'):
    plt.annotate(tissue, (df.Excess.unique()[0], df.neglogq.unique()[0]),
                 fontsize=20)
#     print(df.head())

# plt.axvline(to_plot.FracPosExpected.unique()[0], color='red', ls='--', label='Expected % of Genes UP')
plt.axhline(1, color='black', ls='-.', label='Expected % of Genes UP')
plt.axvline(0, color='red', ls='--')
plt.xlim(-1, 1)
plt.xlabel('Excess Positivity over Expectation')
plt.ylabel('-log$_{10}$(q)')

plt.savefig('../figs/tissue_volcano.svg', bbox_inches='tight', transparent=False)
```

```
/Users/davidangeles/opt/anaconda3/lib/python3.7/site-
packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```



2 PQM-1 tissue analysis

Let's repeat the analysis above, but for our pqm-1 mutant.

[33]:
```
# load stuff:
res = pd.read_csv('../data/master_table.tsv', sep='\t', index_col=0)
# keep DE genes only:
res = res[(res['padj-pqm1'] < 0.05) & (res['Sign-WT'] == 'Same')]

tissues = utils.load_tissues(res.index, 5, 30)
res = res.reindex(tissues.wbid.unique()).dropna()

alpha = 0.05
data = utils.test_tissue_direction(res, tissues=tissues,
                                   col='log2FoldChange-pqm1')
data = utils.fdr_correct(data)
```

```
data.sort_values('fdr', inplace=True)

data['MeanFracPos'] = data.groupby('tissue').FracPos.transform(np.mean)

m = 'There were {0} tissues that changed more in one direction than expected by␣
 ↪random chance in dataset {1}'
print(m.format(len(data[data.sig == True]), n))
```

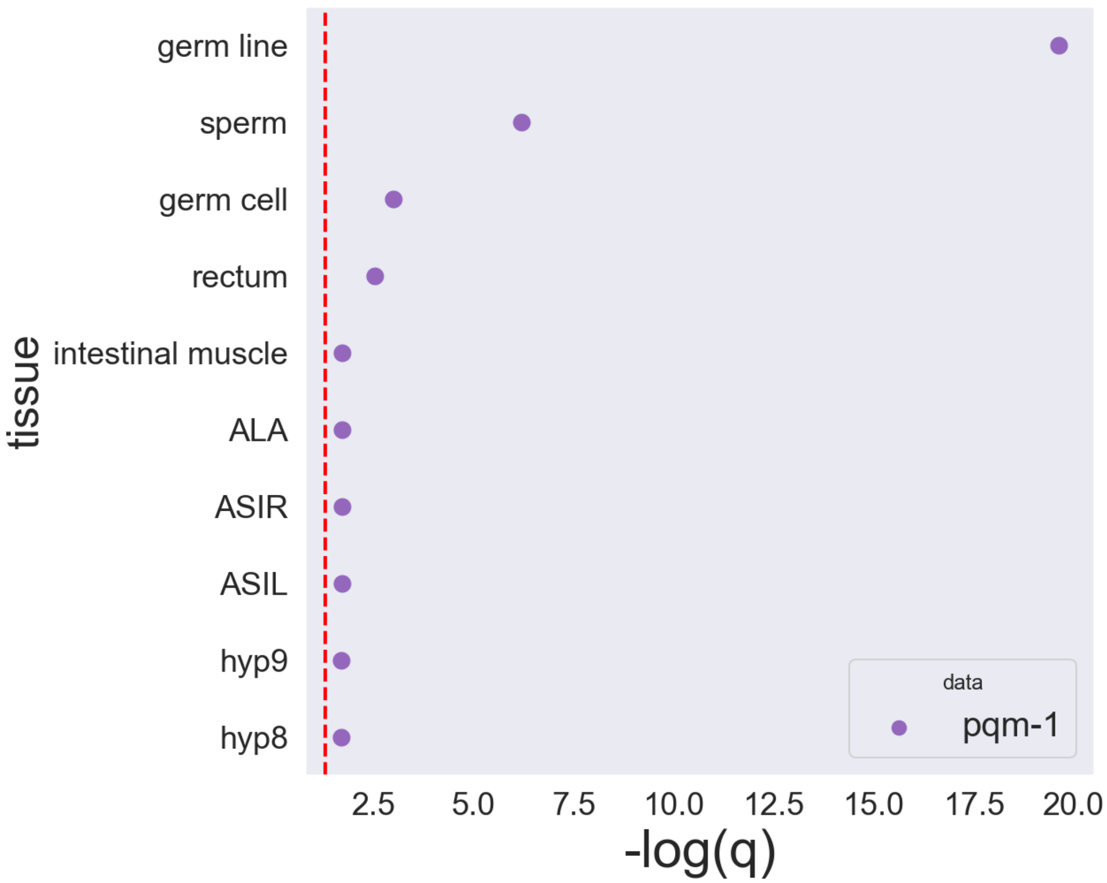
```
tissues originally: 1944
tissues afterwards: 511
There were 19 tissues that changed more in one direction than expected by random
chance in dataset 58hrs
```

[34]:

```
data['data'] = 'pqm-1'
to_plot = data[(~data.tissue.isin(remove)) & (data.sig == True)].head(10)#.
 ↪sort_values('fdr').head(10)
annotated_data = res.join(tissues.set_index('wbid'))
annotated_data['data'] = 'pqm-1'
annotated_data.rename(columns={'log2FoldChange-pqm1':'log2FoldChange'},␣
 ↪inplace=True)

fig, ax = utils.pretty_GSEA_plots(to_plot, annotated_data, alpha, hue='data',␣
 ↪palette={'pqm-1': 'tab:purple'})
ax.set_xlabel('-log(q)', fontsize=30)
```

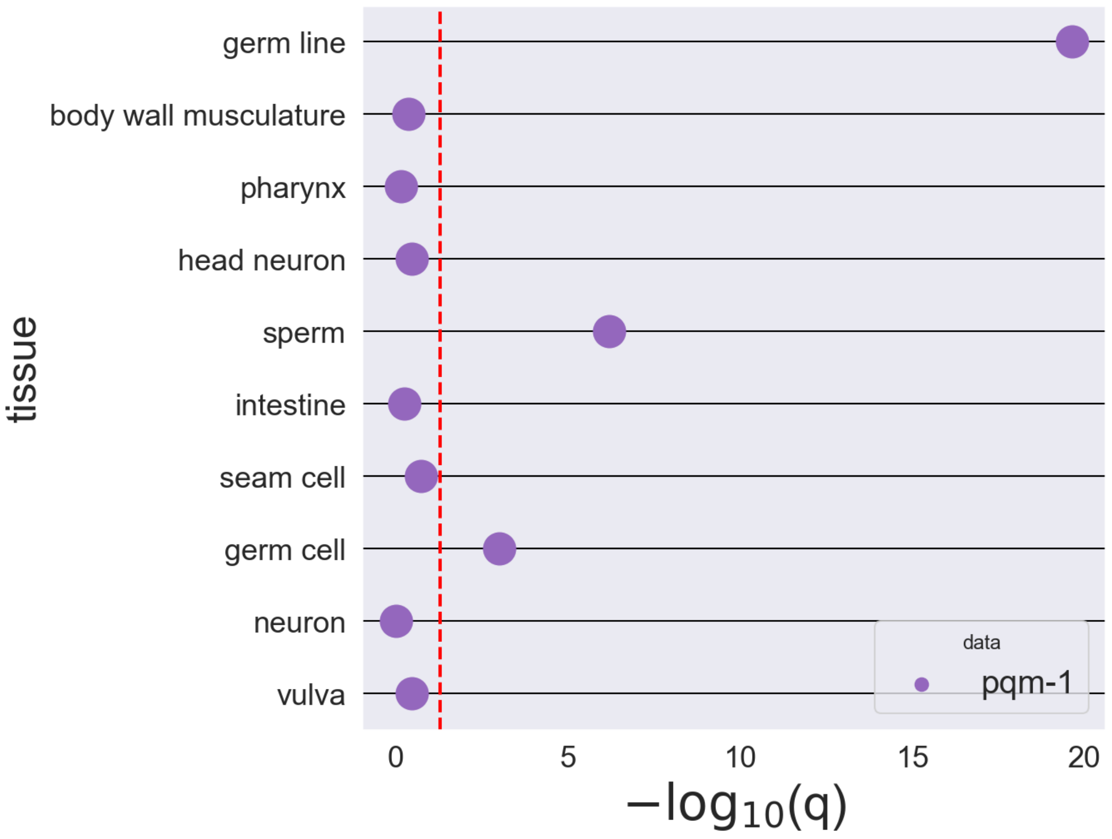
[34]:

```
Text(0.5, 0, '-log(q)')
```



Clearly, the list of tissues is different than for the data at 50 or 58 hours. Let's pull out the tissues we found at 58 hours and plot these to be able to make a comparison:

```
[35]: to_plot = data[(~data.tissue.isin(remove)) & (data.tissue.
       ↪isin(tissues_plotted))].copy()
      to_plot.tissue = to_plot.tissue.astype('category')
      to_plot.tissue.cat.set_categories(tissues_plotted, inplace=True)
      to_plot.sort_values('tissue', inplace=True)

      fig, ax = utils.pretty_GSEA_plots(to_plot, annotated_data, alpha, hue='data',
                                        size=20, palette={'pqm-1': 'tab:purple'})
      ax.set_xlabel('$-\log_{10}$(q)', fontsize=30)
      ax.yaxis.grid(color='black', linewidth=1)
      plt.savefig('../figs/tissue_GSEA_pqm1.svg', bbox_inches='tight',␣
       ↪transparent=False)
```



3 Directional analysis of gene subsets.

In the next section, we pull out all the vitellogenin genes, the major sperm protein genes and all ribosome small / large subunit genes, and we plot their measured log Fold Change at 50 and 58 hours.

[8]:

```
res = pd.read_csv('../data/master_table.tsv', sep='\t', index_col=0)
res = res.dropna(subset=['log2FoldChange-50', 'log2FoldChange-pqm1'])
tissues = utils.load_tissues(res.index, 5, 30)
annotated_data = res.join(tissues.set_index('wbid')).dropna(subset=['tissue'])
```

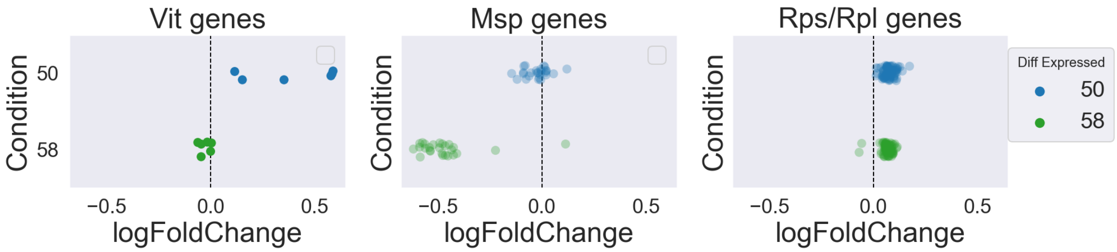
```
tissues originally: 2359
tissues afterwards: 846
```

[41]:

```
def plot_gene_fam(res, title, ax=ax):
    selected = res.Gene.fillna('').str.contains(title)
    colors = {'58': 'tab:green', '50': 'tab:blue', 'pqm1': 'tab:purple'}

    log_cols = ['Gene'] + [c for c in res.columns if ('log2' in c)]
    pval_cols = ['Gene'] + [c for c in res.columns if ('padj' in c)]
    tidy = res[selected][log_cols].melt(id_vars='Gene', var_name='Condition',␣
 ↪value_name='logFoldChange')
    tidy_qvals = res[selected][pval_cols].melt(id_vars='Gene',␣
 ↪var_name='Condition', value_name='qval')

    tidy.Condition = tidy.Condition.str.replace('log2FoldChange-', '')
    tidy_qvals.Condition = tidy_qvals.Condition.str.replace('padj-', '')

    tidy = tidy.join(tidy_qvals.set_index(['Gene', 'Condition']), on=['Gene',␣
 ↪'Condition'])
    tidy['Sig'] = tidy.qval < 0.1

    print(tidy.Gene.unique())
    if len(tidy) > 30:
        alpha = .3
    else:
        alpha=1

    sns.stripplot(y='Condition', x='logFoldChange', hue='Condition',␣
 ↪alpha=alpha,
                  s=8, data=tidy, orient='h', palette=colors, ax=ax)
    ax.axvline(0, color='black', ls='--', lw=1)
    ax.legend(loc=(1, .3), title='Diff Expressed')
    ax.set_title(title.capitalize() + ' genes')


res = pd.read_csv('../data/master_table.tsv', sep='\t', index_col=0)
res = res.dropna(subset=['log2FoldChange-50']).
 ↪drop(columns=['log2FoldChange-pqm1', 'padj-pqm1'])
tissues = utils.load_tissues(res.index, 5, 30)
annotated_data = res.join(tissues.set_index('wbid')).dropna(subset=['tissue'])


fig, ax = plt.subplots(ncols=3, sharey=True, sharex=True, figsize=(15, 2.5))
plot_gene_fam(res, title='vit', ax=ax[0])
plot_gene_fam(res, title='msp', ax=ax[1])
plot_gene_fam(res, title='^rp[sl]', ax=ax[2])

ax[2].set_title('Rps/Rpl genes')

ax[0].legend([])
ax[1].legend([])
plt.savefig('../figs/gene_programs.svg', bbox_inches='tight', transparent=False)
```

```
tissues originally: 2359
tissues afterwards: 846
```

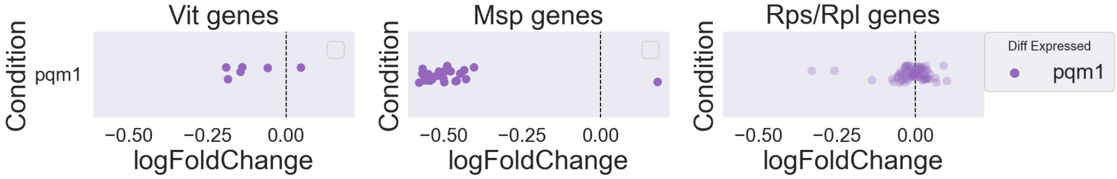
```
['vit-1' 'vit-2' 'vit-3' 'vit-4' 'vit-5' 'vit-6']
['msp-3' 'msp-19' 'msp-31' 'msp-33' 'msp-36' 'msp-38' 'msp-40' 'msp-45'
 'msp-49' 'msp-50' 'msp-51' 'msp-53' 'msp-55' 'msp-56' 'msp-57' 'msp-59'
 'msp-64' 'msp-65' 'msp-76' 'msp-77' 'msp-78' 'msp-81' 'msp-113' 'msp-142'
 'msp-152' 'mspn-1']
['rpl-1' 'rpl-2' 'rpl-3' 'rpl-4' 'rpl-5' 'rpl-6' 'rpl-7' 'rpl-7A' 'rpl-9'
 'rpl-10' 'rpl-11.1' 'rpl-11.2' 'rpl-12' 'rpl-13' 'rpl-14' 'rpl-15'
 'rpl-16' 'rpl-17' 'rpl-18' 'rpl-19' 'rpl-20' 'rpl-21' 'rpl-22' 'rpl-23'
 'rpl-24.1' 'rpl-24.2' 'rpl-25.1' 'rpl-25.2' 'rpl-26' 'rpl-27' 'rpl-28'
 'rpl-29' 'rpl-30' 'rpl-31' 'rpl-32' 'rpl-33' 'rpl-34' 'rpl-35' 'rpl-36'
 'rpl-37' 'rpl-38' 'rpl-39' 'rpl-41' 'rpl-43' 'rps-0' 'rps-1' 'rps-2'
 'rps-3' 'rps-4' 'rps-5' 'rps-6' 'rps-7' 'rps-8' 'rps-9' 'rps-10' 'rps-11'
 'rps-12' 'rps-13' 'rps-14' 'rps-15' 'rps-16' 'rps-17' 'rps-18' 'rps-19'
 'rps-20' 'rps-21' 'rps-22' 'rps-23' 'rps-24' 'rps-25' 'rps-26' 'rps-27'
 'rps-28' 'rps-29' 'rps-30']
```



Next, we repeat the analysis for our pqm-1 mutant data.

[42]:
```
res = pd.read_csv('../data/master_table.tsv', sep='\t', index_col=0)
res = res.dropna(subset=['log2FoldChange-50', 'log2FoldChange-pqm1'])
tissues = utils.load_tissues(res.index, 5, 30)
annotated_data = res.join(tissues.set_index('wbid')).dropna(subset=['tissue'])

res = res.drop(['log2FoldChange-50','log2FoldChange-58'], axis=1)

fig, ax = plt.subplots(ncols=3, sharey=True, sharex=True, figsize=(15, 1.5))
plot_gene_fam(res, title='vit', ax=ax[0])
plot_gene_fam(res, title='msp', ax=ax[1])
plot_gene_fam(res, title='^rp[sl]', ax=ax[2])

ax[2].set_title('Rps/Rpl genes')

ax[0].legend([])
ax[1].legend([])

plt.savefig('../figs/gene_programs_pqm1.svg', bbox_inches='tight')
```

```
tissues originally: 2359
tissues afterwards: 846
['vit-1' 'vit-2' 'vit-3' 'vit-4' 'vit-5' 'vit-6']
['msp-3' 'msp-19' 'msp-31' 'msp-33' 'msp-36' 'msp-38' 'msp-40' 'msp-45'
 'msp-49' 'msp-50' 'msp-51' 'msp-53' 'msp-55' 'msp-56' 'msp-57' 'msp-59'
 'msp-64' 'msp-65' 'msp-76' 'msp-77' 'msp-78' 'msp-81' 'msp-113' 'msp-142'
 'msp-152' 'mspn-1']
['rpl-1' 'rpl-2' 'rpl-3' 'rpl-4' 'rpl-5' 'rpl-6' 'rpl-7' 'rpl-7A' 'rpl-9'
 'rpl-10' 'rpl-11.1' 'rpl-11.2' 'rpl-12' 'rpl-13' 'rpl-14' 'rpl-15'
 'rpl-16' 'rpl-17' 'rpl-18' 'rpl-19' 'rpl-20' 'rpl-21' 'rpl-22' 'rpl-23'
 'rpl-24.1' 'rpl-24.2' 'rpl-25.1' 'rpl-25.2' 'rpl-26' 'rpl-27' 'rpl-28'
 'rpl-29' 'rpl-30' 'rpl-31' 'rpl-32' 'rpl-33' 'rpl-34' 'rpl-35' 'rpl-36'
 'rpl-37' 'rpl-38' 'rpl-39' 'rpl-41' 'rpl-43' 'rps-0' 'rps-1' 'rps-2'
 'rps-3' 'rps-4' 'rps-5' 'rps-6' 'rps-7' 'rps-8' 'rps-9' 'rps-10' 'rps-11'
 'rps-12' 'rps-13' 'rps-14' 'rps-15' 'rps-16' 'rps-17' 'rps-18' 'rps-19'
 'rps-20' 'rps-21' 'rps-22' 'rps-23' 'rps-24' 'rps-25' 'rps-26' 'rps-27'
 'rps-28' 'rps-29' 'rps-30']
```


Vit genes
Msp genes
Rps/Rpl genes
Condition
pqm1
logFoldChange
Diff Expressed
● pqm1