Transcriptomics

January 8, 2021

[8]:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns

from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
from statsmodels.stats.multitest import multipletests
from scipy.stats import hypergeom,norm,zscore,ttest_ind
from boruta import BorutaPy

from tqdm import tqdm_notebook as progbar

sns.set(context='talk',style='ticks',font_scale=1,rc ={'axes.spines.right':␣
 ↪False,'axes.spines.top': False,'pdf.fonttype':42,'ps.fonttype':
 ↪42},palette='colorblind')
```

1 Read in DESeq normalized expression data

[13]:

```
normcounts=pd.read_csv(r'data/Icalvum-DESeq-Norm-Expression.csv',index_col=0)

tdict = {'earlylog':0,'midlog':1,'latelog':2,'stationary':3,'10hours':
 ↪2,'23hours':3}
cols = [x for x in normcounts.columns]
newcols = []
for col in cols:
    x = col.split('.')
    if 'inhibition' in x:
        newcols.append((x[0]+'-inhibition',tdict[x[2]],int(x[3])))
    elif 'injection' in x:
        newcols.append((x[0]+'-injection',tdict[x[3]],int(x[4])))
    else:
        newcols.append((x[0],tdict[x[1]],int(x[2])))
```

```
normcounts = normcounts.loc[:,cols]
normcounts.columns=pd.MultiIndex.from_tuples(newcols)
normcounts=normcounts[['Regular','Cysteine-inhibition','Sulfide-inhibition','Cysteine-injectic

avgnormcounts=normcounts.groupby(level=[0,1],axis=1).mean()
normcounts
```

```
[13]:                    Regular                                              \
                          0                                1
                          1          2          3          1          2
      transcrip_name
      Ga0242637_111       1.764951   1.764951   1.764951   1.764951   1.764951
      Ga0242637_1110      1.764951   2.422089   2.450083   1.764951   1.764951
      Ga0242637_11100     9.647211   9.712475   9.972837   9.113689   9.240558
      Ga0242637_111000    8.964016   8.960565   9.058446   9.328670   9.299505
      Ga0242637_111001    7.780613   7.599761   7.646933   8.455719   8.530247
      …                        …        …        …        …        …
      Ga0242637_11995    12.496901  12.506068  12.638751  12.875452  12.971586
      Ga0242637_11996    14.030684  14.022720  14.287285  14.631815  14.675740
      Ga0242637_11997    12.172248  11.977329  12.306214  12.818020  12.767670
      Ga0242637_11998     4.063438   3.425457   3.099982   3.024689   3.522085
      Ga0242637_11999    11.715580  11.727706  11.782525  11.638385  11.639466

                                                                             …  \
                                     2                                3      …
                          3          1          2          3          1      …
      transcrip_name                                                         …
      Ga0242637_111       1.764951   1.764951   1.764951   1.764951   1.764951  …
      Ga0242637_1110      1.764951   1.764951   1.764951   2.506513   1.764951  …
      Ga0242637_11100     8.934173   7.662034   7.326369   7.402545   6.501520  …
      Ga0242637_111000    9.287764   9.446803   9.308885   9.192409   9.314530  …
      Ga0242637_111001    8.508583   8.462083   8.442209   8.424706   8.314123  …
      …                        …        …        …        …        …  …
      Ga0242637_11995    12.991303  12.830107  12.887992  12.969049  11.997957  …
      Ga0242637_11996    14.678342  14.462971  14.562552  14.668998  13.664627  …
      Ga0242637_11997    12.927888  12.891237  12.913018  13.055429  12.573852  …
      Ga0242637_11998     3.434912   3.155796   3.437259   3.359100   3.165259  …
      Ga0242637_11999    11.563931  11.285909  11.323774  11.354351  11.402249  …

                      Cysteine-injection                                   \
                                       2          3
                                       3          1          2          3
      transcrip_name
      Ga0242637_111             1.764951   1.764951   1.764951   1.764951
      Ga0242637_1110            1.764951   1.764951   1.764951   2.685057
      Ga0242637_11100           8.473396   8.645503   8.697109   8.845633
```

```
Ga0242637_111000          8.845845   8.156454   8.034764   8.068365
Ga0242637_111001          9.762752   9.514957   9.380872   9.605331
…                              …          …          …          …
Ga0242637_11995          11.785669  11.961355  11.952160  12.054452
Ga0242637_11996          13.542052  13.558001  13.688516  13.689068
Ga0242637_11997          11.712794  11.828409  11.747599  11.792460
Ga0242637_11998           2.476742   3.306485   3.617730   3.866372
Ga0242637_11999          10.637587  10.207219  10.418882  10.164699

                 Sulfide-injection                                \
                                3                               2
                                2          3          1         2
transcrip_name
Ga0242637_111            1.764951   1.764951   1.764951  1.764951
Ga0242637_1110           1.764951   1.764951   1.764951  1.764951
Ga0242637_11100          8.654121   7.938396   8.193367  6.309007
Ga0242637_111000         9.760741   9.638828   9.831954  9.171200
Ga0242637_111001         9.954828   9.554752   9.853554  9.203035
…                             …          …          …          …
Ga0242637_11995         13.640810  13.455756  13.504468 12.995306
Ga0242637_11996         15.395846  15.159613  15.283490 14.706892
Ga0242637_11997         13.862552  13.649596  13.845085 12.627090
Ga0242637_11998          3.356064   4.888372   4.896327  3.224071
Ga0242637_11999         11.196452  11.152429  11.191347 11.063698


                         3          1
transcrip_name
Ga0242637_111     1.764951   1.764951
Ga0242637_1110    1.764951   1.764951
Ga0242637_11100   5.737211   6.164488
Ga0242637_111000  8.878948   9.252720
Ga0242637_111001  8.917023   8.885757
…                      …          …
Ga0242637_11995  13.135907  13.099003
Ga0242637_11996  14.787201  14.864108
Ga0242637_11997  12.499114  12.871374
Ga0242637_11998   1.764951   3.570480
Ga0242637_11999  11.206320  11.041612

[3687 rows x 48 columns]
```

2 Create new IDs for use in publication

[14]:

```
gene_map={}
j=1
for i in normcounts.index:
    if i not in gene_map:
        gene_map[i] = 'Ical_%s'%(str(j).zfill(4))
    j+=1
gene_map
```

[14]:

```
{'Ga0242637_111': 'Ical_0001',
 'Ga0242637_1110': 'Ical_0002',
 'Ga0242637_11100': 'Ical_0003',
 'Ga0242637_111000': 'Ical_0004',
 'Ga0242637_111001': 'Ical_0005',
 'Ga0242637_111002': 'Ical_0006',
 'Ga0242637_111003': 'Ical_0007',
 'Ga0242637_111004': 'Ical_0008',
 'Ga0242637_111005': 'Ical_0009',
 'Ga0242637_111006': 'Ical_0010',
 'Ga0242637_111007': 'Ical_0011',
 'Ga0242637_111008': 'Ical_0012',
 'Ga0242637_111009': 'Ical_0013',
 'Ga0242637_11101': 'Ical_0014',
 'Ga0242637_111010': 'Ical_0015',
 'Ga0242637_111015': 'Ical_0016',
 'Ga0242637_111016': 'Ical_0017',
 'Ga0242637_111017': 'Ical_0018',
 'Ga0242637_111018': 'Ical_0019',
 'Ga0242637_111019': 'Ical_0020',
 'Ga0242637_11102': 'Ical_0021',
 'Ga0242637_111020': 'Ical_0022',
 'Ga0242637_111021': 'Ical_0023',
 'Ga0242637_111022': 'Ical_0024',
 'Ga0242637_111023': 'Ical_0025',
 'Ga0242637_111024': 'Ical_0026',
 'Ga0242637_111025': 'Ical_0027',
 'Ga0242637_111026': 'Ical_0028',
 'Ga0242637_111027': 'Ical_0029',
 'Ga0242637_111028': 'Ical_0030',
 'Ga0242637_111029': 'Ical_0031',
 'Ga0242637_11103': 'Ical_0032',
 'Ga0242637_111030': 'Ical_0033',
 'Ga0242637_111031': 'Ical_0034',
 'Ga0242637_111032': 'Ical_0035',
 'Ga0242637_111033': 'Ical_0036',
 'Ga0242637_111034': 'Ical_0037',
 'Ga0242637_111035': 'Ical_0038',
 'Ga0242637_111036': 'Ical_0039',
 'Ga0242637_111037': 'Ical_0040',
 'Ga0242637_111038': 'Ical_0041',
 'Ga0242637_111039': 'Ical_0042',
 'Ga0242637_11104': 'Ical_0043',
 'Ga0242637_111040': 'Ical_0044',
 'Ga0242637_111041': 'Ical_0045',
 'Ga0242637_111042': 'Ical_0046',
 'Ga0242637_111043': 'Ical_0047',
 'Ga0242637_111044': 'Ical_0048',
 'Ga0242637_111045': 'Ical_0049',
 'Ga0242637_111046': 'Ical_0050',
 'Ga0242637_111047': 'Ical_0051',
 'Ga0242637_111048': 'Ical_0052',
 'Ga0242637_111049': 'Ical_0053',
 'Ga0242637_11105': 'Ical_0054',
 'Ga0242637_111050': 'Ical_0055',
 'Ga0242637_111051': 'Ical_0056',
 'Ga0242637_111052': 'Ical_0057',
 'Ga0242637_111053': 'Ical_0058',
 'Ga0242637_111054': 'Ical_0059',
 'Ga0242637_111055': 'Ical_0060',
 'Ga0242637_111056': 'Ical_0061',
 'Ga0242637_111057': 'Ical_0062',
 'Ga0242637_111058': 'Ical_0063',
 'Ga0242637_111059': 'Ical_0064',
 'Ga0242637_11106': 'Ical_0065',
 'Ga0242637_111060': 'Ical_0066',
 'Ga0242637_111061': 'Ical_0067',
 'Ga0242637_111062': 'Ical_0068',
 'Ga0242637_111063': 'Ical_0069',
 'Ga0242637_111064': 'Ical_0070',
 'Ga0242637_111065': 'Ical_0071',
 'Ga0242637_111066': 'Ical_0072',
 'Ga0242637_111067': 'Ical_0073',
 'Ga0242637_111068': 'Ical_0074',
 'Ga0242637_111069': 'Ical_0075',
 'Ga0242637_11107': 'Ical_0076',
 'Ga0242637_111070': 'Ical_0077',
 'Ga0242637_111072': 'Ical_0078',
 'Ga0242637_111073': 'Ical_0079',
 'Ga0242637_111074': 'Ical_0080',
 'Ga0242637_111075': 'Ical_0081',
 'Ga0242637_111076': 'Ical_0082',
 'Ga0242637_111077': 'Ical_0083',
 'Ga0242637_111078': 'Ical_0084',
 'Ga0242637_111079': 'Ical_0085',
 'Ga0242637_11108': 'Ical_0086',
 'Ga0242637_111080': 'Ical_0087',
 'Ga0242637_111081': 'Ical_0088',
 'Ga0242637_111082': 'Ical_0089',
 'Ga0242637_111083': 'Ical_0090',
 'Ga0242637_111084': 'Ical_0091',
 'Ga0242637_111085': 'Ical_0092',
 'Ga0242637_111086': 'Ical_0093',
 'Ga0242637_111087': 'Ical_0094',
 'Ga0242637_111088': 'Ical_0095',
 'Ga0242637_111089': 'Ical_0096',
 'Ga0242637_11109': 'Ical_0097',
 'Ga0242637_111090': 'Ical_0098',
 'Ga0242637_111091': 'Ical_0099',
 'Ga0242637_111092': 'Ical_0100',
 'Ga0242637_111093': 'Ical_0101',
 'Ga0242637_111094': 'Ical_0102',
 'Ga0242637_111095': 'Ical_0103',
 'Ga0242637_111096': 'Ical_0104',
 'Ga0242637_111097': 'Ical_0105',
 'Ga0242637_111098': 'Ical_0106',
 'Ga0242637_111099': 'Ical_0107',
 'Ga0242637_1111': 'Ical_0108',
 'Ga0242637_11110': 'Ical_0109',
 'Ga0242637_111100': 'Ical_0110',
 'Ga0242637_111101': 'Ical_0111',
 'Ga0242637_111102': 'Ical_0112',
 'Ga0242637_111103': 'Ical_0113',
 'Ga0242637_111104': 'Ical_0114',
 'Ga0242637_111106': 'Ical_0115',
 'Ga0242637_111107': 'Ical_0116',
 'Ga0242637_111108': 'Ical_0117',
 'Ga0242637_111109': 'Ical_0118',
 'Ga0242637_11111': 'Ical_0119',
 'Ga0242637_111110': 'Ical_0120',
 'Ga0242637_111111': 'Ical_0121',
 'Ga0242637_111112': 'Ical_0122',
 'Ga0242637_111113': 'Ical_0123',
 'Ga0242637_111114': 'Ical_0124',
 'Ga0242637_111115': 'Ical_0125',
 'Ga0242637_111116': 'Ical_0126',
 'Ga0242637_111117': 'Ical_0127',
 'Ga0242637_111118': 'Ical_0128',
 'Ga0242637_111119': 'Ical_0129',
 'Ga0242637_111120': 'Ical_0130',
 'Ga0242637_111121': 'Ical_0131',
 'Ga0242637_111122': 'Ical_0132',
 'Ga0242637_111123': 'Ical_0133',
 'Ga0242637_111124': 'Ical_0134',
 'Ga0242637_111125': 'Ical_0135',
 'Ga0242637_111126': 'Ical_0136',
 'Ga0242637_111127': 'Ical_0137',
 'Ga0242637_111128': 'Ical_0138',
 'Ga0242637_111129': 'Ical_0139',
 'Ga0242637_11113': 'Ical_0140',
 'Ga0242637_111130': 'Ical_0141',
 'Ga0242637_111131': 'Ical_0142',
 'Ga0242637_111132': 'Ical_0143',
 'Ga0242637_111133': 'Ical_0144',
 'Ga0242637_111134': 'Ical_0145',
 'Ga0242637_111135': 'Ical_0146',
 'Ga0242637_111136': 'Ical_0147',
 'Ga0242637_111138': 'Ical_0148',
 'Ga0242637_111139': 'Ical_0149',
 'Ga0242637_11114': 'Ical_0150',
 'Ga0242637_111140': 'Ical_0151',
 'Ga0242637_111142': 'Ical_0152',
 'Ga0242637_111144': 'Ical_0153',
 'Ga0242637_111145': 'Ical_0154',
 'Ga0242637_111147': 'Ical_0155',
 'Ga0242637_111148': 'Ical_0156',
 'Ga0242637_111149': 'Ical_0157',
 'Ga0242637_11115': 'Ical_0158',
 'Ga0242637_111150': 'Ical_0159',
 'Ga0242637_111151': 'Ical_0160',
 'Ga0242637_111152': 'Ical_0161',
 'Ga0242637_111153': 'Ical_0162',
 'Ga0242637_111154': 'Ical_0163',
 'Ga0242637_111155': 'Ical_0164',
 'Ga0242637_111156': 'Ical_0165',
 'Ga0242637_111157': 'Ical_0166',
 'Ga0242637_111158': 'Ical_0167',
 'Ga0242637_111159': 'Ical_0168',
 'Ga0242637_11116': 'Ical_0169',
 'Ga0242637_111160': 'Ical_0170',
 'Ga0242637_111161': 'Ical_0171',
 'Ga0242637_111162': 'Ical_0172',
 'Ga0242637_111163': 'Ical_0173',
 'Ga0242637_111164': 'Ical_0174',
 'Ga0242637_111165': 'Ical_0175',
 'Ga0242637_111166': 'Ical_0176',
 'Ga0242637_111167': 'Ical_0177',
 'Ga0242637_111168': 'Ical_0178',
 'Ga0242637_111169': 'Ical_0179',
 'Ga0242637_111170': 'Ical_0180',
 'Ga0242637_111171': 'Ical_0181',
 'Ga0242637_111172': 'Ical_0182',
 'Ga0242637_111173': 'Ical_0183',
 'Ga0242637_111174': 'Ical_0184',
 'Ga0242637_111175': 'Ical_0185',
 'Ga0242637_111176': 'Ical_0186',
 'Ga0242637_111179': 'Ical_0187',
 'Ga0242637_11118': 'Ical_0188',
 'Ga0242637_111180': 'Ical_0189',
 'Ga0242637_111181': 'Ical_0190',
 'Ga0242637_111182': 'Ical_0191',
 'Ga0242637_111183': 'Ical_0192',
 'Ga0242637_111184': 'Ical_0193',
 'Ga0242637_111185': 'Ical_0194',
 'Ga0242637_111186': 'Ical_0195',
 'Ga0242637_111187': 'Ical_0196',
 'Ga0242637_111188': 'Ical_0197',
 'Ga0242637_111189': 'Ical_0198',
 'Ga0242637_11119': 'Ical_0199',
 'Ga0242637_111190': 'Ical_0200',
 'Ga0242637_111191': 'Ical_0201',
 'Ga0242637_111192': 'Ical_0202',
 'Ga0242637_111193': 'Ical_0203',
 'Ga0242637_111194': 'Ical_0204',
 'Ga0242637_111195': 'Ical_0205',
 'Ga0242637_111196': 'Ical_0206',
 'Ga0242637_111197': 'Ical_0207',
 'Ga0242637_111198': 'Ical_0208',
 'Ga0242637_111199': 'Ical_0209',
 'Ga0242637_1112': 'Ical_0210',
 'Ga0242637_111200': 'Ical_0211',
 'Ga0242637_111201': 'Ical_0212',
 'Ga0242637_111202': 'Ical_0213',
 'Ga0242637_111203': 'Ical_0214',
 'Ga0242637_111204': 'Ical_0215',
 'Ga0242637_111205': 'Ical_0216',
 'Ga0242637_111206': 'Ical_0217',
 'Ga0242637_111207': 'Ical_0218',
 'Ga0242637_111208': 'Ical_0219',
 'Ga0242637_111209': 'Ical_0220',
 'Ga0242637_11121': 'Ical_0221',
 'Ga0242637_111210': 'Ical_0222',
 'Ga0242637_111211': 'Ical_0223',
 'Ga0242637_111212': 'Ical_0224',
 'Ga0242637_111213': 'Ical_0225',
 'Ga0242637_111214': 'Ical_0226',
 'Ga0242637_111215': 'Ical_0227',
 'Ga0242637_111216': 'Ical_0228',
 'Ga0242637_111217': 'Ical_0229',
 'Ga0242637_111219': 'Ical_0230',
 'Ga0242637_11122': 'Ical_0231',
 'Ga0242637_111220': 'Ical_0232',
 'Ga0242637_111221': 'Ical_0233',
 'Ga0242637_111222': 'Ical_0234',
 'Ga0242637_111223': 'Ical_0235',
 'Ga0242637_111224': 'Ical_0236',
 'Ga0242637_111225': 'Ical_0237',
 'Ga0242637_111227': 'Ical_0238',
 'Ga0242637_111228': 'Ical_0239',
 'Ga0242637_111229': 'Ical_0240',
 'Ga0242637_11123': 'Ical_0241',
 'Ga0242637_111230': 'Ical_0242',
 'Ga0242637_111231': 'Ical_0243',
 'Ga0242637_111232': 'Ical_0244',
 'Ga0242637_111233': 'Ical_0245',
 'Ga0242637_111234': 'Ical_0246',
 'Ga0242637_111235': 'Ical_0247',
 'Ga0242637_111236': 'Ical_0248',
 'Ga0242637_111237': 'Ical_0249',
 'Ga0242637_111238': 'Ical_0250',
 'Ga0242637_111239': 'Ical_0251',
 'Ga0242637_11124': 'Ical_0252',
 'Ga0242637_111240': 'Ical_0253',
 'Ga0242637_111241': 'Ical_0254',
 'Ga0242637_111242': 'Ical_0255',
 'Ga0242637_111243': 'Ical_0256',
 'Ga0242637_111244': 'Ical_0257',
 'Ga0242637_111245': 'Ical_0258',
 'Ga0242637_111246': 'Ical_0259',
 'Ga0242637_111247': 'Ical_0260',
 'Ga0242637_111248': 'Ical_0261',
 'Ga0242637_111249': 'Ical_0262',
 'Ga0242637_11125': 'Ical_0263',
 'Ga0242637_111250': 'Ical_0264',
 'Ga0242637_111251': 'Ical_0265',
 'Ga0242637_111252': 'Ical_0266',
 'Ga0242637_111253': 'Ical_0267',
 'Ga0242637_111254': 'Ical_0268',
 'Ga0242637_111255': 'Ical_0269',
 'Ga0242637_111256': 'Ical_0270',
 'Ga0242637_111257': 'Ical_0271',
 'Ga0242637_111258': 'Ical_0272',
 'Ga0242637_111259': 'Ical_0273',
 'Ga0242637_11126': 'Ical_0274',
 'Ga0242637_111260': 'Ical_0275',
 'Ga0242637_111261': 'Ical_0276',
 'Ga0242637_111262': 'Ical_0277',
 'Ga0242637_111263': 'Ical_0278',
 'Ga0242637_111264': 'Ical_0279',
 'Ga0242637_111265': 'Ical_0280',
 'Ga0242637_111266': 'Ical_0281',
 'Ga0242637_111267': 'Ical_0282',
 'Ga0242637_111268': 'Ical_0283',
 'Ga0242637_111269': 'Ical_0284',
 'Ga0242637_11127': 'Ical_0285',
 'Ga0242637_111270': 'Ical_0286',
 'Ga0242637_111271': 'Ical_0287',
 'Ga0242637_111272': 'Ical_0288',
 'Ga0242637_111273': 'Ical_0289',
 'Ga0242637_111274': 'Ical_0290',
 'Ga0242637_111275': 'Ical_0291',
 'Ga0242637_111276': 'Ical_0292',
 'Ga0242637_111277': 'Ical_0293',
 'Ga0242637_111278': 'Ical_0294',
 'Ga0242637_111279': 'Ical_0295',
 'Ga0242637_11128': 'Ical_0296',
 'Ga0242637_111280': 'Ical_0297',
 'Ga0242637_111281': 'Ical_0298',
 'Ga0242637_111282': 'Ical_0299',
 'Ga0242637_111283': 'Ical_0300',
 'Ga0242637_111284': 'Ical_0301',
 'Ga0242637_111285': 'Ical_0302',
 'Ga0242637_111286': 'Ical_0303',
 'Ga0242637_111287': 'Ical_0304',
 'Ga0242637_111288': 'Ical_0305',
 'Ga0242637_111289': 'Ical_0306',
 'Ga0242637_11129': 'Ical_0307',
 'Ga0242637_111290': 'Ical_0308',
 'Ga0242637_111291': 'Ical_0309',
 'Ga0242637_111292': 'Ical_0310',
 'Ga0242637_111293': 'Ical_0311',
 'Ga0242637_111294': 'Ical_0312',
 'Ga0242637_111295': 'Ical_0313',
 'Ga0242637_111296': 'Ical_0314',
 'Ga0242637_111297': 'Ical_0315',
 'Ga0242637_111298': 'Ical_0316',
 'Ga0242637_111299': 'Ical_0317',
 'Ga0242637_1113': 'Ical_0318',
 'Ga0242637_11130': 'Ical_0319',
 'Ga0242637_111300': 'Ical_0320',
 'Ga0242637_111301': 'Ical_0321',
 'Ga0242637_111302': 'Ical_0322',
 'Ga0242637_111303': 'Ical_0323',
 'Ga0242637_111304': 'Ical_0324',
 'Ga0242637_111305': 'Ical_0325',
 'Ga0242637_111306': 'Ical_0326',
 'Ga0242637_111307': 'Ical_0327',
 'Ga0242637_111308': 'Ical_0328',
 'Ga0242637_111309': 'Ical_0329',
 'Ga0242637_11131': 'Ical_0330',
 'Ga0242637_111310': 'Ical_0331',
 'Ga0242637_111311': 'Ical_0332',
 'Ga0242637_111312': 'Ical_0333',
 'Ga0242637_111313': 'Ical_0334',
 'Ga0242637_111314': 'Ical_0335',
 'Ga0242637_111315': 'Ical_0336',
 'Ga0242637_111316': 'Ical_0337',
 'Ga0242637_111318': 'Ical_0338',
 'Ga0242637_111319': 'Ical_0339',
 'Ga0242637_11132': 'Ical_0340',
 'Ga0242637_111320': 'Ical_0341',
 'Ga0242637_111321': 'Ical_0342',
 'Ga0242637_111322': 'Ical_0343',
 'Ga0242637_111323': 'Ical_0344',
 'Ga0242637_111324': 'Ical_0345',
 'Ga0242637_111325': 'Ical_0346',
 'Ga0242637_111326': 'Ical_0347',
 'Ga0242637_111327': 'Ical_0348',
 'Ga0242637_111328': 'Ical_0349',
 'Ga0242637_111329': 'Ical_0350',
 'Ga0242637_11133': 'Ical_0351',
 'Ga0242637_111330': 'Ical_0352',
 'Ga0242637_111331': 'Ical_0353',
 'Ga0242637_111332': 'Ical_0354',
 'Ga0242637_111333': 'Ical_0355',
 'Ga0242637_111334': 'Ical_0356',
 'Ga0242637_111335': 'Ical_0357',
 'Ga0242637_111336': 'Ical_0358',
 'Ga0242637_111337': 'Ical_0359',
 'Ga0242637_111338': 'Ical_0360',
 'Ga0242637_111339': 'Ical_0361',
 'Ga0242637_11134': 'Ical_0362',
 'Ga0242637_111340': 'Ical_0363',
 'Ga0242637_111341': 'Ical_0364',
 'Ga0242637_111342': 'Ical_0365',
 'Ga0242637_111343': 'Ical_0366',
 'Ga0242637_111344': 'Ical_0367',
 'Ga0242637_111345': 'Ical_0368',
 'Ga0242637_111346': 'Ical_0369',
 'Ga0242637_111347': 'Ical_0370',
 'Ga0242637_111348': 'Ical_0371',
 'Ga0242637_111349': 'Ical_0372',
 'Ga0242637_11135': 'Ical_0373',
 'Ga0242637_111350': 'Ical_0374',
 'Ga0242637_111351': 'Ical_0375',
 'Ga0242637_111352': 'Ical_0376',
 'Ga0242637_111353': 'Ical_0377',
 'Ga0242637_111354': 'Ical_0378',
 'Ga0242637_111355': 'Ical_0379',
 'Ga0242637_111356': 'Ical_0380',
 'Ga0242637_111357': 'Ical_0381',
 'Ga0242637_111358': 'Ical_0382',
 'Ga0242637_111359': 'Ical_0383',
 'Ga0242637_11136': 'Ical_0384',
 'Ga0242637_111360': 'Ical_0385',
 'Ga0242637_111361': 'Ical_0386',
 'Ga0242637_111362': 'Ical_0387',
 'Ga0242637_111363': 'Ical_0388',
 'Ga0242637_111364': 'Ical_0389',
 'Ga0242637_111365': 'Ical_0390',
 'Ga0242637_111366': 'Ical_0391',
 'Ga0242637_111367': 'Ical_0392',
 'Ga0242637_111368': 'Ical_0393',
 'Ga0242637_111369': 'Ical_0394',
 'Ga0242637_11137': 'Ical_0395',
 'Ga0242637_111370': 'Ical_0396',
 'Ga0242637_111371': 'Ical_0397',
 'Ga0242637_111372': 'Ical_0398',
 'Ga0242637_111373': 'Ical_0399',
 'Ga0242637_111374': 'Ical_0400',
 'Ga0242637_111375': 'Ical_0401',
 'Ga0242637_111376': 'Ical_0402',
 'Ga0242637_111377': 'Ical_0403',
 'Ga0242637_11138': 'Ical_0404',
 'Ga0242637_111382': 'Ical_0405',
 'Ga0242637_111383': 'Ical_0406',
 'Ga0242637_111384': 'Ical_0407',
 'Ga0242637_111385': 'Ical_0408',
 'Ga0242637_111386': 'Ical_0409',
 'Ga0242637_111387': 'Ical_0410',
 'Ga0242637_111388': 'Ical_0411',
 'Ga0242637_111389': 'Ical_0412',
 'Ga0242637_11139': 'Ical_0413',
 'Ga0242637_111390': 'Ical_0414',
 'Ga0242637_111391': 'Ical_0415',
 'Ga0242637_111392': 'Ical_0416',
 'Ga0242637_111393': 'Ical_0417',
 'Ga0242637_111394': 'Ical_0418',
 'Ga0242637_111395': 'Ical_0419',
 'Ga0242637_111396': 'Ical_0420',
 'Ga0242637_111397': 'Ical_0421',
 'Ga0242637_111398': 'Ical_0422',
 'Ga0242637_111399': 'Ical_0423',
 'Ga0242637_1114': 'Ical_0424',
 'Ga0242637_11140': 'Ical_0425',
 'Ga0242637_111400': 'Ical_0426',
 'Ga0242637_111401': 'Ical_0427',
 'Ga0242637_111402': 'Ical_0428',
 'Ga0242637_111403': 'Ical_0429',
 'Ga0242637_111404': 'Ical_0430',
 'Ga0242637_111405': 'Ical_0431',
 'Ga0242637_111406': 'Ical_0432',
 'Ga0242637_111407': 'Ical_0433',
 'Ga0242637_111408': 'Ical_0434',
 'Ga0242637_111409': 'Ical_0435',
 'Ga0242637_11141': 'Ical_0436',
 'Ga0242637_111410': 'Ical_0437',
 'Ga0242637_111411': 'Ical_0438',
 'Ga0242637_111412': 'Ical_0439',
 'Ga0242637_111413': 'Ical_0440',
 'Ga0242637_111414': 'Ical_0441',
 'Ga0242637_111415': 'Ical_0442',
 'Ga0242637_111416': 'Ical_0443',
 'Ga0242637_111417': 'Ical_0444',
 'Ga0242637_111418': 'Ical_0445',
 'Ga0242637_111419': 'Ical_0446',
 'Ga0242637_11142': 'Ical_0447',
 'Ga0242637_111420': 'Ical_0448',
 'Ga0242637_111421': 'Ical_0449',
 'Ga0242637_111422': 'Ical_0450',
 'Ga0242637_111423': 'Ical_0451',
 'Ga0242637_111424': 'Ical_0452',
 'Ga0242637_111425': 'Ical_0453',
 'Ga0242637_111426': 'Ical_0454',
 'Ga0242637_111427': 'Ical_0455',
 'Ga0242637_111428': 'Ical_0456',
 'Ga0242637_111429': 'Ical_0457',
 'Ga0242637_11143': 'Ical_0458',
 'Ga0242637_111430': 'Ical_0459',
 'Ga0242637_111431': 'Ical_0460',
 'Ga0242637_111432': 'Ical_0461',
 'Ga0242637_111433': 'Ical_0462',
 'Ga0242637_111434': 'Ical_0463',
 'Ga0242637_111435': 'Ical_0464',
 'Ga0242637_111436': 'Ical_0465',
 'Ga0242637_111437': 'Ical_0466',
 'Ga0242637_111438': 'Ical_0467',
 'Ga0242637_111439': 'Ical_0468',
 'Ga0242637_11144': 'Ical_0469',
 'Ga0242637_111441': 'Ical_0470',
 'Ga0242637_111442': 'Ical_0471',
 'Ga0242637_111443': 'Ical_0472',
 'Ga0242637_111444': 'Ical_0473',
 'Ga0242637_111445': 'Ical_0474',
 'Ga0242637_111446': 'Ical_0475',
 'Ga0242637_111447': 'Ical_0476',
 'Ga0242637_111448': 'Ical_0477',
 'Ga0242637_111449': 'Ical_0478',
 'Ga0242637_11145': 'Ical_0479',
 'Ga0242637_111450': 'Ical_0480',
 'Ga0242637_111451': 'Ical_0481',
 'Ga0242637_111452': 'Ical_0482',
 'Ga0242637_111453': 'Ical_0483',
 'Ga0242637_111454': 'Ical_0484',
 'Ga0242637_111455': 'Ical_0485',
 'Ga0242637_111456': 'Ical_0486',
 'Ga0242637_111457': 'Ical_0487',
 'Ga0242637_111458': 'Ical_0488',
 'Ga0242637_111459': 'Ical_0489',
 'Ga0242637_11146': 'Ical_0490',
 'Ga0242637_111460': 'Ical_0491',
 'Ga0242637_111461': 'Ical_0492',
 'Ga0242637_111462': 'Ical_0493',
 'Ga0242637_111463': 'Ical_0494',
 'Ga0242637_111464': 'Ical_0495',
 'Ga0242637_111465': 'Ical_0496',
 'Ga0242637_111466': 'Ical_0497',
 'Ga0242637_111467': 'Ical_0498',
 'Ga0242637_111469': 'Ical_0499',
 'Ga0242637_11147': 'Ical_0500',
 'Ga0242637_111470': 'Ical_0501',
 'Ga0242637_111471': 'Ical_0502',
 'Ga0242637_111472': 'Ical_0503',
 'Ga0242637_111473': 'Ical_0504',
 'Ga0242637_111474': 'Ical_0505',
 'Ga0242637_111475': 'Ical_0506',
 'Ga0242637_111476': 'Ical_0507',
 'Ga0242637_111477': 'Ical_0508',
 'Ga0242637_111478': 'Ical_0509',
 'Ga0242637_111479': 'Ical_0510',
 'Ga0242637_11148': 'Ical_0511',
 'Ga0242637_111480': 'Ical_0512',
 'Ga0242637_111481': 'Ical_0513',
 'Ga0242637_111482': 'Ical_0514',
 'Ga0242637_111483': 'Ical_0515',
 'Ga0242637_111484': 'Ical_0516',
 'Ga0242637_111485': 'Ical_0517',
 'Ga0242637_111486': 'Ical_0518',
 'Ga0242637_111487': 'Ical_0519',
 'Ga0242637_111488': 'Ical_0520',
 'Ga0242637_111489': 'Ical_0521',
 'Ga0242637_11149': 'Ical_0522',
 'Ga0242637_111490': 'Ical_0523',
 'Ga0242637_111491': 'Ical_0524',
 'Ga0242637_111492': 'Ical_0525',
 'Ga0242637_111493': 'Ical_0526',
 'Ga0242637_111494': 'Ical_0527',
 'Ga0242637_111495': 'Ical_0528',
 'Ga0242637_111496': 'Ical_0529',
 'Ga0242637_111497': 'Ical_0530',
 'Ga0242637_111498': 'Ical_0531',
 'Ga0242637_111499': 'Ical_0532',
 'Ga0242637_1115': 'Ical_0533',
 'Ga0242637_11150': 'Ical_0534',
 'Ga0242637_111500': 'Ical_0535',
 'Ga0242637_111501': 'Ical_0536',
 'Ga0242637_111502': 'Ical_0537',
 'Ga0242637_111503': 'Ical_0538',
 'Ga0242637_111504': 'Ical_0539',
 'Ga0242637_111505': 'Ical_0540',
 'Ga0242637_111506': 'Ical_0541',
 'Ga0242637_111507': 'Ical_0542',
 'Ga0242637_111508': 'Ical_0543',
 'Ga0242637_111509': 'Ical_0544',
 'Ga0242637_11151': 'Ical_0545',
 'Ga0242637_111510': 'Ical_0546',
 'Ga0242637_111511': 'Ical_0547',
 'Ga0242637_111512': 'Ical_0548',
 'Ga0242637_111513': 'Ical_0549',
 'Ga0242637_111514': 'Ical_0550',
 'Ga0242637_111515': 'Ical_0551',
 'Ga0242637_111516': 'Ical_0552',
 'Ga0242637_111517': 'Ical_0553',
 'Ga0242637_111518': 'Ical_0554',
 'Ga0242637_111519': 'Ical_0555',
 'Ga0242637_11152': 'Ical_0556',
 'Ga0242637_111520': 'Ical_0557',
 'Ga0242637_111521': 'Ical_0558',
 'Ga0242637_111522': 'Ical_0559',
 'Ga0242637_111523': 'Ical_0560',
 'Ga0242637_111524': 'Ical_0561',
 'Ga0242637_111525': 'Ical_0562',
 'Ga0242637_111526': 'Ical_0563',
 'Ga0242637_111527': 'Ical_0564',
 'Ga0242637_111528': 'Ical_0565',
 'Ga0242637_111529': 'Ical_0566',
 'Ga0242637_11153': 'Ical_0567',
 'Ga0242637_111530': 'Ical_0568',
 'Ga0242637_111531': 'Ical_0569',
 'Ga0242637_111532': 'Ical_0570',
 'Ga0242637_111533': 'Ical_0571',
 'Ga0242637_111534': 'Ical_0572',
 'Ga0242637_111535': 'Ical_0573',
 'Ga0242637_111536': 'Ical_0574',
 'Ga0242637_111537': 'Ical_0575',
 'Ga0242637_111538': 'Ical_0576',
 'Ga0242637_111539': 'Ical_0577',
 'Ga0242637_11154': 'Ical_0578',
 'Ga0242637_111540': 'Ical_0579',
 'Ga0242637_111541': 'Ical_0580',
 'Ga0242637_111542': 'Ical_0581',
 'Ga0242637_111543': 'Ical_0582',
 'Ga0242637_111544': 'Ical_0583',
 'Ga0242637_111545': 'Ical_0584',
 'Ga0242637_111546': 'Ical_0585',
 'Ga0242637_111547': 'Ical_0586',
 'Ga0242637_111548': 'Ical_0587',
 'Ga0242637_111549': 'Ical_0588',
 'Ga0242637_11155': 'Ical_0589',
 'Ga0242637_111550': 'Ical_0590',
 'Ga0242637_111551': 'Ical_0591',
 'Ga0242637_111552': 'Ical_0592',
 'Ga0242637_111553': 'Ical_0593',
 'Ga0242637_111554': 'Ical_0594',
 'Ga0242637_111555': 'Ical_0595',
 'Ga0242637_111556': 'Ical_0596',
 'Ga0242637_111557': 'Ical_0597',
 'Ga0242637_111558': 'Ical_0598',
 'Ga0242637_111559': 'Ical_0599',
 'Ga0242637_11156': 'Ical_0600',
 'Ga0242637_111560': 'Ical_0601',
 'Ga0242637_111561': 'Ical_0602',
 'Ga0242637_111562': 'Ical_0603',
 'Ga0242637_111563': 'Ical_0604',
 'Ga0242637_111564': 'Ical_0605',
 'Ga0242637_111566': 'Ical_0606',
 'Ga0242637_111567': 'Ical_0607',
 'Ga0242637_111568': 'Ical_0608',
 'Ga0242637_111569': 'Ical_0609',
 'Ga0242637_11157': 'Ical_0610',
 'Ga0242637_111570': 'Ical_0611',
 'Ga0242637_111571': 'Ical_0612',
 'Ga0242637_111572': 'Ical_0613',
 'Ga0242637_111573': 'Ical_0614',
 'Ga0242637_111574': 'Ical_0615',
 'Ga0242637_111575': 'Ical_0616',
 'Ga0242637_111576': 'Ical_0617',
 'Ga0242637_111577': 'Ical_0618',
 'Ga0242637_111578': 'Ical_0619',
 'Ga0242637_111579': 'Ical_0620',
 'Ga0242637_11158': 'Ical_0621',
 'Ga0242637_111580': 'Ical_0622',
 'Ga0242637_111581': 'Ical_0623',
 'Ga0242637_111582': 'Ical_0624',
 'Ga0242637_111583': 'Ical_0625',
 'Ga0242637_111584': 'Ical_0626',
 'Ga0242637_111585': 'Ical_0627',
 'Ga0242637_111586': 'Ical_0628',
 'Ga0242637_111587': 'Ical_0629',
 'Ga0242637_111588': 'Ical_0630',
 'Ga0242637_111589': 'Ical_0631',
 'Ga0242637_11159': 'Ical_0632',
 'Ga0242637_111590': 'Ical_0633',
 'Ga0242637_111591': 'Ical_0634',
 'Ga0242637_111592': 'Ical_0635',
 'Ga0242637_111593': 'Ical_0636',
 'Ga0242637_111594': 'Ical_0637',
 'Ga0242637_111595': 'Ical_0638',
 'Ga0242637_111596': 'Ical_0639',
 'Ga0242637_111597': 'Ical_0640',
 'Ga0242637_111598': 'Ical_0641',
 'Ga0242637_111599': 'Ical_0642',
 'Ga0242637_1116': 'Ical_0643',
 'Ga0242637_11160': 'Ical_0644',
 'Ga0242637_111600': 'Ical_0645',
 'Ga0242637_111601': 'Ical_0646',
 'Ga0242637_111602': 'Ical_0647',
 'Ga0242637_111603': 'Ical_0648',
 'Ga0242637_111604': 'Ical_0649',
 'Ga0242637_111605': 'Ical_0650',
 'Ga0242637_111606': 'Ical_0651',
 'Ga0242637_111607': 'Ical_0652',
 'Ga0242637_111608': 'Ical_0653',
 'Ga0242637_111609': 'Ical_0654',
 'Ga0242637_11161': 'Ical_0655',
 'Ga0242637_111610': 'Ical_0656',
 'Ga0242637_111611': 'Ical_0657',
 'Ga0242637_111612': 'Ical_0658',
 'Ga0242637_111613': 'Ical_0659',
 'Ga0242637_111614': 'Ical_0660',
 'Ga0242637_111615': 'Ical_0661',
 'Ga0242637_111616': 'Ical_0662',
 'Ga0242637_111617': 'Ical_0663',
 'Ga0242637_111618': 'Ical_0664',
 'Ga0242637_111619': 'Ical_0665',
 'Ga0242637_11162': 'Ical_0666',
 'Ga0242637_111620': 'Ical_0667',
 'Ga0242637_111621': 'Ical_0668',
 'Ga0242637_111622': 'Ical_0669',
 'Ga0242637_111623': 'Ical_0670',
 'Ga0242637_111624': 'Ical_0671',
 'Ga0242637_111625': 'Ical_0672',
 'Ga0242637_111626': 'Ical_0673',
 'Ga0242637_111627': 'Ical_0674',
 'Ga0242637_111628': 'Ical_0675',
 'Ga0242637_111629': 'Ical_0676',
 'Ga0242637_11163': 'Ical_0677',
 'Ga0242637_111630': 'Ical_0678',
 'Ga0242637_111631': 'Ical_0679',
 'Ga0242637_111632': 'Ical_0680',
 'Ga0242637_111633': 'Ical_0681',
 'Ga0242637_111634': 'Ical_0682',
 'Ga0242637_111635': 'Ical_0683',
 'Ga0242637_111636': 'Ical_0684',
 'Ga0242637_111637': 'Ical_0685',
 'Ga0242637_111638': 'Ical_0686',
 'Ga0242637_111639': 'Ical_0687',
 'Ga0242637_11164': 'Ical_0688',
 'Ga0242637_111641': 'Ical_0689',
 'Ga0242637_111642': 'Ical_0690',
 'Ga0242637_111643': 'Ical_0691',
 'Ga0242637_111644': 'Ical_0692',
 'Ga0242637_111645': 'Ical_0693',
 'Ga0242637_111646': 'Ical_0694',
 'Ga0242637_111647': 'Ical_0695',
 'Ga0242637_111648': 'Ical_0696',
 'Ga0242637_111649': 'Ical_0697',
 'Ga0242637_11165': 'Ical_0698',
 'Ga0242637_111650': 'Ical_0699',
 'Ga0242637_111651': 'Ical_0700',
 'Ga0242637_111652': 'Ical_0701',
 'Ga0242637_111653': 'Ical_0702',
 'Ga0242637_111654': 'Ical_0703',
 'Ga0242637_111655': 'Ical_0704',
 'Ga0242637_111656': 'Ical_0705',
 'Ga0242637_111657': 'Ical_0706',
 'Ga0242637_111659': 'Ical_0707',
 'Ga0242637_11166': 'Ical_0708',
 'Ga0242637_111660': 'Ical_0709',
 'Ga0242637_111661': 'Ical_0710',
 'Ga0242637_111662': 'Ical_0711',
 'Ga0242637_111663': 'Ical_0712',
 'Ga0242637_111664': 'Ical_0713',
 'Ga0242637_111665': 'Ical_0714',
 'Ga0242637_111666': 'Ical_0715',
 'Ga0242637_111667': 'Ical_0716',
 'Ga0242637_111668': 'Ical_0717',
 'Ga0242637_111669': 'Ical_0718',
 'Ga0242637_11167': 'Ical_0719',
 'Ga0242637_111670': 'Ical_0720',
 'Ga0242637_111671': 'Ical_0721',
 'Ga0242637_111672': 'Ical_0722',
 'Ga0242637_111673': 'Ical_0723',
 'Ga0242637_111674': 'Ical_0724',
 'Ga0242637_111675': 'Ical_0725',
 'Ga0242637_111676': 'Ical_0726',
 'Ga0242637_111677': 'Ical_0727',
 'Ga0242637_111678': 'Ical_0728',
 'Ga0242637_111679': 'Ical_0729',
 'Ga0242637_11168': 'Ical_0730',
 'Ga0242637_111680': 'Ical_0731',
 'Ga0242637_111681': 'Ical_0732',
 'Ga0242637_111682': 'Ical_0733',
 'Ga0242637_111683': 'Ical_0734',
 'Ga0242637_111684': 'Ical_0735',
 'Ga0242637_111685': 'Ical_0736',
 'Ga0242637_111686': 'Ical_0737',
 'Ga0242637_111687': 'Ical_0738',
 'Ga0242637_111688': 'Ical_0739',
 'Ga0242637_111689': 'Ical_0740',
 'Ga0242637_11169': 'Ical_0741',
 'Ga0242637_111690': 'Ical_0742',
 'Ga0242637_111691': 'Ical_0743',
 'Ga0242637_111692': 'Ical_0744',
 'Ga0242637_111693': 'Ical_0745',
 'Ga0242637_111694': 'Ical_0746',
 'Ga0242637_111695': 'Ical_0747',
 'Ga0242637_111696': 'Ical_0748',
 'Ga0242637_111697': 'Ical_0749',
 'Ga0242637_111698': 'Ical_0750',
 'Ga0242637_111699': 'Ical_0751',
 'Ga0242637_1117': 'Ical_0752',
 'Ga0242637_11170': 'Ical_0753',
 'Ga0242637_111700': 'Ical_0754',
 'Ga0242637_111701': 'Ical_0755',
 'Ga0242637_111702': 'Ical_0756',
 'Ga0242637_111703': 'Ical_0757',
 'Ga0242637_111704': 'Ical_0758',
 'Ga0242637_111705': 'Ical_0759',
 'Ga0242637_111706': 'Ical_0760',
 'Ga0242637_111707': 'Ical_0761',
 'Ga0242637_111708': 'Ical_0762',
 'Ga0242637_111709': 'Ical_0763',
 'Ga0242637_11171': 'Ical_0764',
 'Ga0242637_111710': 'Ical_0765',
 'Ga0242637_111711': 'Ical_0766',
 'Ga0242637_111712': 'Ical_0767',
 'Ga0242637_111713': 'Ical_0768',
 'Ga0242637_111714': 'Ical_0769',
 'Ga0242637_111715': 'Ical_0770',
 'Ga0242637_111716': 'Ical_0771',
 'Ga0242637_111717': 'Ical_0772',
 'Ga0242637_111718': 'Ical_0773',
 'Ga0242637_111719': 'Ical_0774',
 'Ga0242637_11172': 'Ical_0775',
 'Ga0242637_111720': 'Ical_0776',
 'Ga0242637_111721': 'Ical_0777',
 'Ga0242637_111723': 'Ical_0778',
 'Ga0242637_111724': 'Ical_0779',
 'Ga0242637_111725': 'Ical_0780',
 'Ga0242637_111726': 'Ical_0781',
 'Ga0242637_111727': 'Ical_0782',
 'Ga0242637_111728': 'Ical_0783',
 'Ga0242637_11173': 'Ical_0784',
 'Ga0242637_111733': 'Ical_0785',
 'Ga0242637_111734': 'Ical_0786',
 'Ga0242637_111735': 'Ical_0787',
 'Ga0242637_111736': 'Ical_0788',
 'Ga0242637_111737': 'Ical_0789',
 'Ga0242637_111738': 'Ical_0790',
 'Ga0242637_111739': 'Ical_0791',
 'Ga0242637_11174': 'Ical_0792',
 'Ga0242637_111740': 'Ical_0793',
 'Ga0242637_111741': 'Ical_0794',
 'Ga0242637_111742': 'Ical_0795',
 'Ga0242637_111743': 'Ical_0796',
 'Ga0242637_111744': 'Ical_0797',
 'Ga0242637_111745': 'Ical_0798',
 'Ga0242637_111746': 'Ical_0799',
 'Ga0242637_111747': 'Ical_0800',
 'Ga0242637_111748': 'Ical_0801',
 'Ga0242637_111749': 'Ical_0802',
 'Ga0242637_11175': 'Ical_0803',
 'Ga0242637_111750': 'Ical_0804',
 'Ga0242637_111751': 'Ical_0805',
 'Ga0242637_111752': 'Ical_0806',
 'Ga0242637_111753': 'Ical_0807',
 'Ga0242637_111754': 'Ical_0808',
 'Ga0242637_111755': 'Ical_0809',
 'Ga0242637_111756': 'Ical_0810',
 'Ga0242637_111757': 'Ical_0811',
 'Ga0242637_111758': 'Ical_0812',
 'Ga0242637_111759': 'Ical_0813',
 'Ga0242637_11176': 'Ical_0814',
 'Ga0242637_111760': 'Ical_0815',
 'Ga0242637_111761': 'Ical_0816',
 'Ga0242637_111762': 'Ical_0817',
 'Ga0242637_111763': 'Ical_0818',
 'Ga0242637_111764': 'Ical_0819',
 'Ga0242637_111765': 'Ical_0820',
 'Ga0242637_111766': 'Ical_0821',
 'Ga0242637_111767': 'Ical_0822',
 'Ga0242637_111768': 'Ical_0823',
 'Ga0242637_111769': 'Ical_0824',
 'Ga0242637_11177': 'Ical_0825',
 'Ga0242637_111770': 'Ical_0826',
 'Ga0242637_111771': 'Ical_0827',
 'Ga0242637_111772': 'Ical_0828',
 'Ga0242637_111773': 'Ical_0829',
 'Ga0242637_111774': 'Ical_0830',
 'Ga0242637_111775': 'Ical_0831',
 'Ga0242637_111776': 'Ical_0832',
 'Ga0242637_111777': 'Ical_0833',
 'Ga0242637_111778': 'Ical_0834',
 'Ga0242637_111779': 'Ical_0835',
 'Ga0242637_11178': 'Ical_0836',
 'Ga0242637_111780': 'Ical_0837',
 'Ga0242637_111781': 'Ical_0838',
 'Ga0242637_111782': 'Ical_0839',
 'Ga0242637_111783': 'Ical_0840',
 'Ga0242637_111784': 'Ical_0841',
 'Ga0242637_111785': 'Ical_0842',
 'Ga0242637_111786': 'Ical_0843',
 'Ga0242637_111787': 'Ical_0844',
 'Ga0242637_111788': 'Ical_0845',
 'Ga0242637_111789': 'Ical_0846',
 'Ga0242637_11179': 'Ical_0847',
 'Ga0242637_111790': 'Ical_0848',
 'Ga0242637_111791': 'Ical_0849',
 'Ga0242637_111792': 'Ical_0850',
 'Ga0242637_111793': 'Ical_0851',
 'Ga0242637_111794': 'Ical_0852',
 'Ga0242637_111795': 'Ical_0853',
 'Ga0242637_111796': 'Ical_0854',
 'Ga0242637_111797': 'Ical_0855',
 'Ga0242637_111798': 'Ical_0856',
 'Ga0242637_111799': 'Ical_0857',
 'Ga0242637_1118': 'Ical_0858',
 'Ga0242637_11180': 'Ical_0859',
 'Ga0242637_111800': 'Ical_0860',
 'Ga0242637_111801': 'Ical_0861',
 'Ga0242637_111802': 'Ical_0862',
 'Ga0242637_111803': 'Ical_0863',
 'Ga0242637_111804': 'Ical_0864',
 'Ga0242637_111805': 'Ical_0865',
 'Ga0242637_111806': 'Ical_0866',
 'Ga0242637_111807': 'Ical_0867',
 'Ga0242637_111808': 'Ical_0868',
 'Ga0242637_111809': 'Ical_0869',
 'Ga0242637_11181': 'Ical_0870',
 'Ga0242637_111810': 'Ical_0871',
 'Ga0242637_111811': 'Ical_0872',
 'Ga0242637_111812': 'Ical_0873',
 'Ga0242637_111813': 'Ical_0874',
 'Ga0242637_111814': 'Ical_0875',
 'Ga0242637_111815': 'Ical_0876',
 'Ga0242637_111816': 'Ical_0877',
 'Ga0242637_111817': 'Ical_0878',
 'Ga0242637_111818': 'Ical_0879',
 'Ga0242637_111819': 'Ical_0880',
 'Ga0242637_11182': 'Ical_0881',
 'Ga0242637_111820': 'Ical_0882',
 'Ga0242637_111821': 'Ical_0883',
 'Ga0242637_111822': 'Ical_0884',
 'Ga0242637_111823': 'Ical_0885',
 'Ga0242637_111824': 'Ical_0886',
 'Ga0242637_111825': 'Ical_0887',
 'Ga0242637_111826': 'Ical_0888',
 'Ga0242637_111827': 'Ical_0889',
 'Ga0242637_111828': 'Ical_0890',
 'Ga0242637_111829': 'Ical_0891',
 'Ga0242637_11183': 'Ical_0892',
 'Ga0242637_111830': 'Ical_0893',
 'Ga0242637_111831': 'Ical_0894',
 'Ga0242637_111832': 'Ical_0895',
 'Ga0242637_111833': 'Ical_0896',
 'Ga0242637_111834': 'Ical_0897',
 'Ga0242637_111835': 'Ical_0898',
 'Ga0242637_111836': 'Ical_0899',
 'Ga0242637_111837': 'Ical_0900',
 'Ga0242637_111838': 'Ical_0901',
 'Ga0242637_111839': 'Ical_0902',
 'Ga0242637_11184': 'Ical_0903',
 'Ga0242637_111840': 'Ical_0904',
 'Ga0242637_111841': 'Ical_0905',
 'Ga0242637_111842': 'Ical_0906',
 'Ga0242637_111843': 'Ical_0907',
 'Ga0242637_111844': 'Ical_0908',
 'Ga0242637_111845': 'Ical_0909',
 'Ga0242637_111846': 'Ical_0910',
 'Ga0242637_111847': 'Ical_0911',
 'Ga0242637_111848': 'Ical_0912',
 'Ga0242637_111849': 'Ical_0913',
 'Ga0242637_11185': 'Ical_0914',
 'Ga0242637_111850': 'Ical_0915',
 'Ga0242637_111851': 'Ical_0916',
 'Ga0242637_111852': 'Ical_0917',
 'Ga0242637_111853': 'Ical_0918',
 'Ga0242637_111854': 'Ical_0919',
 'Ga0242637_111855': 'Ical_0920',
 'Ga0242637_111856': 'Ical_0921',
 'Ga0242637_111857': 'Ical_0922',
 'Ga0242637_111858': 'Ical_0923',
 'Ga0242637_111859': 'Ical_0924',
 'Ga0242637_11186': 'Ical_0925',
 'Ga0242637_111860': 'Ical_0926',
 'Ga0242637_111861': 'Ical_0927',
 'Ga0242637_111862': 'Ical_0928',
 'Ga0242637_111863': 'Ical_0929',
 'Ga0242637_111864': 'Ical_0930',
 'Ga0242637_111865': 'Ical_0931',
 'Ga0242637_111866': 'Ical_0932',
 'Ga0242637_111867': 'Ical_0933',
 'Ga0242637_111868': 'Ical_0934',
 'Ga0242637_111869': 'Ical_0935',
 'Ga0242637_11187': 'Ical_0936',
 'Ga0242637_111870': 'Ical_0937',
 'Ga0242637_111871': 'Ical_0938',
 'Ga0242637_111872': 'Ical_0939',
 'Ga0242637_111873': 'Ical_0940',
 'Ga0242637_111874': 'Ical_0941',
 'Ga0242637_111875': 'Ical_0942',
 'Ga0242637_111876': 'Ical_0943',
 'Ga0242637_111878': 'Ical_0944',
 'Ga0242637_111879': 'Ical_0945',
 'Ga0242637_11188': 'Ical_0946',
 'Ga0242637_111880': 'Ical_0947',
 'Ga0242637_111881': 'Ical_0948',
 'Ga0242637_111882': 'Ical_0949',
 'Ga0242637_111883': 'Ical_0950',
 'Ga0242637_111884': 'Ical_0951',
 'Ga0242637_111885': 'Ical_0952',
 'Ga0242637_111887': 'Ical_0953',
 'Ga0242637_111888': 'Ical_0954',
 'Ga0242637_111889': 'Ical_0955',
 'Ga0242637_11189': 'Ical_0956',
 'Ga0242637_111890': 'Ical_0957',
 'Ga0242637_111891': 'Ical_0958',
 'Ga0242637_111892': 'Ical_0959',
 'Ga0242637_111893': 'Ical_0960',
 'Ga0242637_111894': 'Ical_0961',
 'Ga0242637_111895': 'Ical_0962',
 'Ga0242637_111896': 'Ical_0963',
 'Ga0242637_111897': 'Ical_0964',
 'Ga0242637_111898': 'Ical_0965',
 'Ga0242637_111899': 'Ical_0966',
 'Ga0242637_1119': 'Ical_0967',
 'Ga0242637_11190': 'Ical_0968',
 'Ga0242637_111900': 'Ical_0969',
 'Ga0242637_111901': 'Ical_0970',
 'Ga0242637_111902': 'Ical_0971',
 'Ga0242637_111903': 'Ical_0972',
 'Ga0242637_111904': 'Ical_0973',
 'Ga0242637_111905': 'Ical_0974',
 'Ga0242637_111906': 'Ical_0975',
 'Ga0242637_111907': 'Ical_0976',
 'Ga0242637_111909': 'Ical_0977',
 'Ga0242637_11191': 'Ical_0978',
 'Ga0242637_111910': 'Ical_0979',
 'Ga0242637_111911': 'Ical_0980',
 'Ga0242637_111912': 'Ical_0981',
 'Ga0242637_111913': 'Ical_0982',
 'Ga0242637_111914': 'Ical_0983',
 'Ga0242637_111915': 'Ical_0984',
 'Ga0242637_111916': 'Ical_0985',
 'Ga0242637_111917': 'Ical_0986',
 'Ga0242637_111918': 'Ical_0987',
 'Ga0242637_111919': 'Ical_0988',
 'Ga0242637_11192': 'Ical_0989',
 'Ga0242637_111920': 'Ical_0990',
 'Ga0242637_111921': 'Ical_0991',
 'Ga0242637_111922': 'Ical_0992',
 'Ga0242637_111923': 'Ical_0993',
 'Ga0242637_111924': 'Ical_0994',
 'Ga0242637_111925': 'Ical_0995',
 'Ga0242637_111926': 'Ical_0996',
 'Ga0242637_111927': 'Ical_0997',
 'Ga0242637_111928': 'Ical_0998',
 'Ga0242637_111929': 'Ical_0999',
 'Ga0242637_11193': 'Ical_1000',
 …}
```

3 Read in gene annotations

- These annotations are obtained by BLAST to match transcript sequences to a database of annotations from other bacteria

3.0.1 SEED Annotations

- V2 is percent match (similarity of sequences) of Icalvum transcript to annotation hit (from database)
- V3 is p-value of match
- Aready filtered to remove hits that were not significant

3.0.2 GO Annotations

- Similar to SEED but use different annotation scheme, as with SEED these are obtained by looking for annotated genes that match ours

3.0.3 MetaCyc

- Pathway centric, have correspondance with XCMS online analysis

```
[15]: seed = pd.read_csv(r'data/seed_annotations.csv',index_col=0)
      seed
```

[15]:

```
                                      superpathway         pathway  \
query
Ga0242637_116                         unclassified    unclassified
Ga0242637_117                         unclassified    unclassified
Ga0242637_118              Experimental Subsystems    unclassified
Ga0242637_1110                     Stress Response  Osmotic stress
Ga0242637_1111                     Stress Response  Osmotic stress
…                                            …             …
Ga0242637_113749  Metabolism of Aromatic Compounds    unclassified
Ga0242637_113752                      unclassified    unclassified
Ga0242637_113753                      unclassified    unclassified
Ga0242637_113754                    DNA Metabolism      DNA repair
Ga0242637_113755                    DNA Metabolism      DNA repair

                                                         subpathway  \
query
Ga0242637_116                                                  Yggs
Ga0242637_117                                                  Yggs
Ga0242637_118                              uridine kinase cluster 1
Ga0242637_1110    Choline and Betaine Uptake and Betaine Biosynt…
Ga0242637_1111    Choline and Betaine Uptake and Betaine Biosynt…
…                                                             …
Ga0242637_113749                           Aromatic Amin Catabolism
Ga0242637_113752                                            Que-2 2
Ga0242637_113753                                            Que-2 2
Ga0242637_113754                              DNA repair, bacterial
Ga0242637_113755                              DNA repair, bacterial

                                                           function       id  \
query
Ga0242637_116              Cell division protein FtsH (EC 3.4.24.-)  SS16788
Ga0242637_117              Cell division protein FtsH (EC 3.4.24.-)  SS16788
Ga0242637_118     Deoxycytidine triphosphate deaminase (EC 3.5.4… SS10675
Ga0242637_1110    L-proline glycine betaine ABC transport system… SS02981
Ga0242637_1111    L-proline glycine betaine ABC transport system… SS02982
…                                                             …      …
Ga0242637_113749          Aldehyde dehydrogenase (EC 1.2.1.3), PaaZ  SS16339
Ga0242637_113752                                   Exonuclease SbcC  SS04537
Ga0242637_113753                                   Exonuclease SbcC  SS04537
Ga0242637_113754                                   Exonuclease SbcD  SS04538
Ga0242637_113755                                   Exonuclease SbcD  SS04538

                                                        description  \
query
Ga0242637_116              Cell division protein FtsH (EC 3.4.24.-)
Ga0242637_117              Cell division protein FtsH (EC 3.4.24.-)
Ga0242637_118     Deoxycytidine triphosphate deaminase (EC 3.5.4…
Ga0242637_1110    L-proline glycine betaine ABC transport system…
Ga0242637_1111    L-proline glycine betaine ABC transport system…
…                                                               …
Ga0242637_113749          Aldehyde dehydrogenase (EC 1.2.1.3), PaaZ
Ga0242637_113752                                   Exonuclease SbcC
Ga0242637_113753                                   Exonuclease SbcC
Ga0242637_113754                                   Exonuclease SbcD
Ga0242637_113755                                   Exonuclease SbcD

                         V2             V3
query
Ga0242637_116     85.900000   9.500000e-11
Ga0242637_117     53.083333   4.330000e-20
Ga0242637_118     88.900000   4.000000e-88
Ga0242637_1110    67.542857   2.800000e-57
Ga0242637_1111    72.733333   3.870000e-57
…                       …             …
Ga0242637_113749  59.200000  1.600000e-175
Ga0242637_113752  47.500000   2.070000e-43
Ga0242637_113753  46.886667   3.204242e-02
Ga0242637_113754  55.580000   6.920000e-16
Ga0242637_113755  64.700000   7.000000e-19

[2367 rows x 8 columns]
```

[596]:

```
go_terms = pd.read_csv(r'data/GO-Annotations.csv')
biol=go_terms[go_terms['type']=='GO terms (biological process)']
go_terms
```

[596]:

```
                     ID                                               value  \
0       Ga0242637_11473                                   DNA recombination
1       Ga0242637_11473                                          DNA repair
2       Ga0242637_11473                                        DNA ligation
3       Ga0242637_11473                                     ligase activity
4       Ga0242637_11473                                         ATP binding
…                   …                                                   …
26815  Ga0242637_112258  hydrolase activity, hydrolyzing O-glycosyl com…
26816  Ga0242637_112258         1,4-alpha-glucan branching enzyme activity
26817  Ga0242637_112258                                  catalytic activity
26818  Ga0242637_112258       maltose alpha-D-glucosyltransferase activity
26819  Ga0242637_112258  transferase activity, transferring hexosyl groups

                               type
0      GO terms (biological process)
1      GO terms (biological process)
2      GO terms (biological process)
3      GO terms (molecular function)
```

```
4      GO terms (molecular function)
…                             …
26815  GO terms (molecular function)
26816  GO terms (molecular function)
26817  GO terms (molecular function)
26818  GO terms (molecular function)
26819  GO terms (molecular function)

[26820 rows x 3 columns]
```

[17]:
```python
path_annot = pd.read_csv(r'data/Ical_MetaCyc_Pathways.csv')
path_annot
```

[17]:
```
              intca                ID  \
0     INTCA_RS12115  Ga0242637_112246
1     INTCA_RS12115  Ga0242637_112246
2     INTCA_RS12115  Ga0242637_112246
3     INTCA_RS12115  Ga0242637_112246
4     INTCA_RS12115  Ga0242637_112246
…               …               …
4828  INTCA_RS11855  Ga0242637_112193
4829  INTCA_RS11855  Ga0242637_112193
4830  INTCA_RS16840  Ga0242637_113185
4831  INTCA_RS16840  Ga0242637_113185
4832  INTCA_RS08620  Ga0242637_111525

                                                  value            type
0     <i>N</i><sup>10</sup>-formyl-tetrahydrofolate …         pathway
1                 L-methionine biosynthesis II (plants)         pathway
2               <i>S</i>-adenosyl-L-methionine cycle II         pathway
3                         L-methionine biosynthesis III         pathway
4                <i>S</i>-adenosyl-L-methionine cycle I         pathway
…                                                 …              …
4828                              Allantoin Degradation  parent-pathway
4829                                      Superpathways  parent-pathway
4830                              Allantoin Degradation  parent-pathway
4831                                      Superpathways  parent-pathway
4832                   Phosphatidylcholine Biosynthesis  parent-pathway

[4833 rows x 4 columns]
```

[18]:
```python
# General set of annotations that map Icalvum C5 and Icalvum KIP7 features
# Allows the use of MetaCyc and GO annotations which are KIP7 specific
blast_annot = pd.read_csv(r'data/C5.annotation.v4.txt',index_col=0,sep='\t')
blast_annot
```

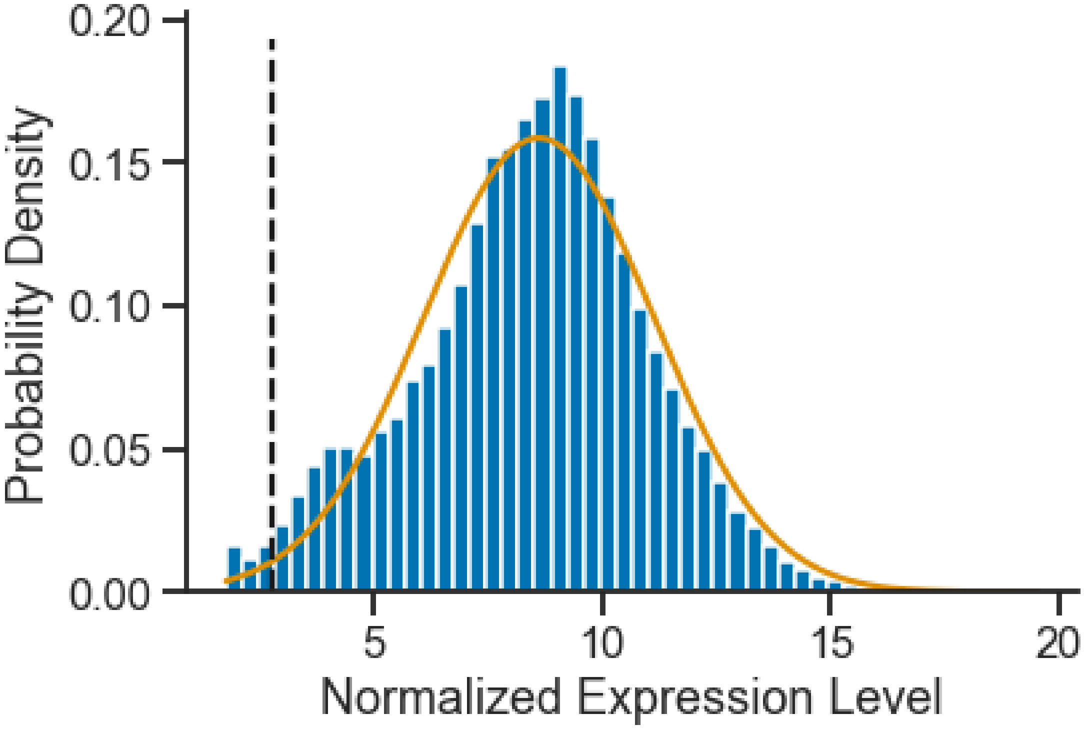
[18]:

```
                     C5.transcript.ID  \
#C5.transcript.name
Ga0242637_111              2769691124
Ga0242637_112              2769691125
Ga0242637_113              2769691126
Ga0242637_114              2769691127
Ga0242637_115              2769691128
…                                   …
Ga0242637_113751           2769694865
Ga0242637_113752           2769694866
Ga0242637_113753           2769694867
Ga0242637_113754           2769694868
Ga0242637_113755           2769694869

                                         C5.transcript.description  \
#C5.transcript.name
Ga0242637_111                                 hypothetical protein
Ga0242637_112                            transcription factor WhiB
Ga0242637_113                       site-specific recombinase XerD
Ga0242637_114                                 hypothetical protein
Ga0242637_115                                                  NaN
…                                                            …
Ga0242637_113751         putative cell wall binding repeat protein
Ga0242637_113752                              exonuclease SbcC
Ga0242637_113753                              hypothetical protein
Ga0242637_113754     type 5 capsule protein repressor-like protein
Ga0242637_113755                              hypothetical protein

                    KIP7.ortholog.MO KIP7.ortholog.RefSeq  \
#C5.transcript.name
Ga0242637_111
Ga0242637_112
Ga0242637_113
Ga0242637_114
Ga0242637_115
…                              …                  …
Ga0242637_113751            11067141        INTCA_RS01210
Ga0242637_113752
Ga0242637_113753
Ga0242637_113754
Ga0242637_113755

                    KIP7.ortholog.RefSeq.old.ID         GO       KEGG  \
#C5.transcript.name
Ga0242637_111
Ga0242637_112                                                pfam02467
Ga0242637_113                                                pfam00589
```

```
Ga0242637_114
Ga0242637_115
…                                         …        …        …
Ga0242637_113751                     Intca_0240        NaN
Ga0242637_113752                                 KO:K03546  pfam13558
Ga0242637_113753
Ga0242637_113754                                            pfam12320
Ga0242637_113755

                          PFAM COG TIGR
#C5.transcript.name
Ga0242637_111                       NaN
Ga0242637_112                       NaN
Ga0242637_113          COG4974      NaN
Ga0242637_114                       NaN
Ga0242637_115                       NaN
…                        …  ..  …
Ga0242637_113751     pfam04122
Ga0242637_113752       COG0419      NaN
Ga0242637_113753                    NaN
Ga0242637_113754                    NaN
Ga0242637_113755                    NaN

[3746 rows x 10 columns]
```

[19]:

```python
#Extract values of normalized counts as an 1-D array
vals = avgnormcounts.values.ravel()
#Create Histogram with density on y-axis (height of all bars sum to 1)
hist = plt.hist(vals,bins=50,density=True)
#Fit normal distribution using counts
dist = norm(loc=np.median(vals),scale=np.std(vals))
#Plot fit of distribution over the histogram
plt.plot(sorted(vals),dist.pdf(sorted(vals)))
plt.xlabel('Normalized Expression Level')
plt.ylabel('Probability Density')
#Filter out genes that are lowly expressed in more than 75% of all samples and have no annotation
keep = []
toss = []
#Set low expression threshold, which in this case is lower 5% of expression distribution
cutoff = 0.01
#Obtain the expression value that corresponds to lower 5% of distribution
x = dist.ppf(cutoff)
#Plot cut-off as vertical dashed line
plt.plot((x,x),plt.ylim(),'k--')
for gene in avgnormcounts.index:
    vals = avgnormcounts.loc[gene,:].values
    truth = dist.cdf(vals)>cutoff
    #Looks at proportion of genes with value == True (e.g. above cutoff)
    #If gene is above cutoff in 25% or more of samples and has an annotation␣
↪maybe we'll keep it
    if len(truth[truth==True])>0 and (gene in seed.index or gene in go_terms.ID.
↪values):
        try:
            name = seed.loc[gene,'function']
        except:
            name = go_terms[go_terms.ID==gene]
            name=name[name.Type=='GO terms (biological process)']['Value']
        #If there are more than one annotation just take the first one (would␣
↪probably be better to take one with lowest p-value)
        if type(name) != type(''):
            if name.shape[0] > 0:
                name = name.iloc[0]
            else:
                name='unclassified'
        #If annotation is hypothetical or unclassified etc. then toss it
        if 'hypothetical' not in name and 'unclassified' not in name and␣
↪'putative' not in name:
            keep.append(gene)
    else:
        toss.append(gene)
```

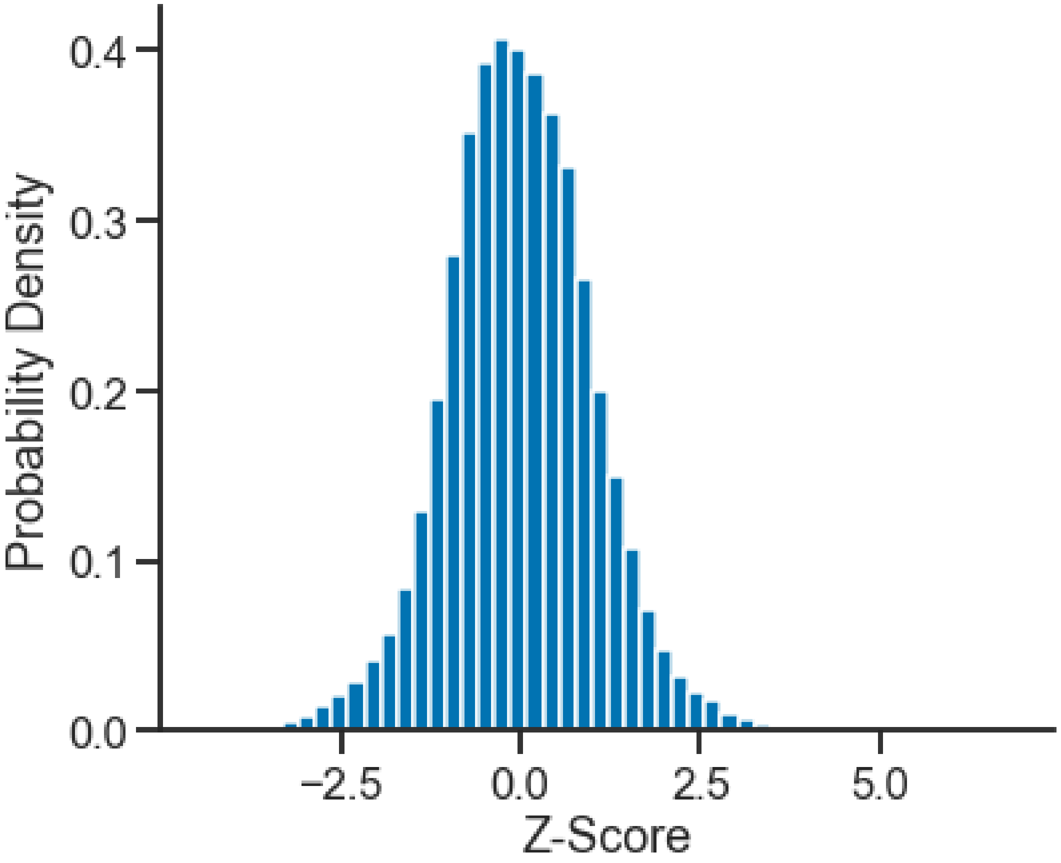


[239]:

```python
#Format and further normalize gene expression data for clustering and other␣
↪analyses
#Additional normalization shifts mean of expression level for each gene across␣
↪conditions to 0 and scales by standard deviation
#This is needed for clustering and PCA, so trends are captured and not␣
↪confounded by absolute value of expression level
#Only is use genes that passed threshold if desired, here that is not done␣
↪because Boruta will narrow our results anyway
avgnorm = normcounts.loc[:].groupby(level=[0,1],axis=1).mean()
avgzvals=avgnorm.T.apply(zscore).T
zvals=normcounts.loc[:,:].T.apply(zscore).T.dropna()
vals = zvals.values.ravel()
plt.figure(figsize=(6,5))
hist = plt.hist(vals,bins=50,density=True)
plt.xlabel('Z-Score')
plt.ylabel('Probability Density')
```

[239]:

```
Text(0, 0.5, 'Probability Density')
```



4 Identify set of genes which best explain differences in conditions

- Uses BORUTA, which is a feature selection algorithm, and the random forest classifier
- Perform all pairwise condition comparisons to identify genes that explain differences in each pair, for each comparison all replicates and time points are used
- Uses scaled counts (max, min, mean normalized) to reduce impact of genes with really high expression
- Can be a bit slow (~5-10 mins)

[275]:

```
#Get hits using Boruta
sets ={}
hits = []
pairs=[]
cond=['Regular','Cysteine-inhibition','Sulfide-inhibition']
for i in cond:
    for j in cond:
        if i != j:
            if [i,j] not in pairs and [j,i] not in pairs:
                pairs.append([i,j])
for pair in progbar(pairs):
    df = zvals.loc[:,pair]
    df = df.T.reset_index()
    if 'injection' in pair[1]:
        df=df[(df['level_1']==2)|(df['level_1']==3)]
    y = df['level_0'].values
    X = df.iloc[:,3:].values
    # define random forest classifier, with utilising all cores and
    # sampling in proportion to y labels
    rf = RandomForestClassifier(n_jobs=-1)

    # define Boruta feature selection method
    feat_selector = BorutaPy(rf, n_estimators='auto', random_state=1,perc=90)

    # find all relevant features
    feat_selector.fit(X, y)

    # call transform() on X to filter it down to selected features
    X_filtered = feat_selector.transform(X)
    sets[tuple(pair)] = list(df.iloc[:,3:].columns[feat_selector.support_])
    hits += sets[tuple(pair)]
#Get rid of duplicate hits
hits=pd.unique(hits)

#Boruta results
```

```
for key in sets.keys():
    print(key,len(sets[key]))
print(len(hits))
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  # Remove the CWD from sys.path while we load stuff.

HBox(children=(FloatProgress(value=0.0, max=3.0), HTML(value='')))
```

5 Principle Component Analysis (PCA)

- Can be useful to visualize differences in conditions based on data
- Like Boruta, which used random forest, PCA is a way of reducing dimensionality of data and identifying differences
- PCA is a linear method, it finds linear combinations of the features (genes) which best explain differences in conditions
- PC1 (e.g. x) = w1g1+w2g2+w3g3..... where w1 is the weight (or importance of feature) and g1 is the feature (gene) value. Thus 1000 genes can be combined into one variable.
- Below 2 dimensions are used for ease of visualization. Conditions are colored, but time points are not distinguished.
- In general PC1 explains most variance, followed by PC2, then PC3 etc…

```
[18]: #All genes
      cond=['Regular','Sulfide-inhibition','Cysteine-inhibition']
      pca_data =zvals.loc[:,cond].copy().T
      pca = PCA(n_components=2)
      pca_df = pd.DataFrame(pca.
       ↪fit_transform(pca_data),columns=['PC1','PC2'],index=pca_data.index)
      pca_df=pca_df.reset_index()
      pca_df.columns=['Condition','Time','Rep','PC1','PC2']
      pca_df['Time']=['T%s'%i for i in pca_df['Time']+1]
      print('Explained Variance... PC1:%s, PC2:%s'%tuple(pca.
       ↪explained_variance_ratio_))
      pca_weights = pd.DataFrame(pca.components_,index=['PC1','PC2'],columns=pca_data.
       ↪columns)
      pca_weights
```

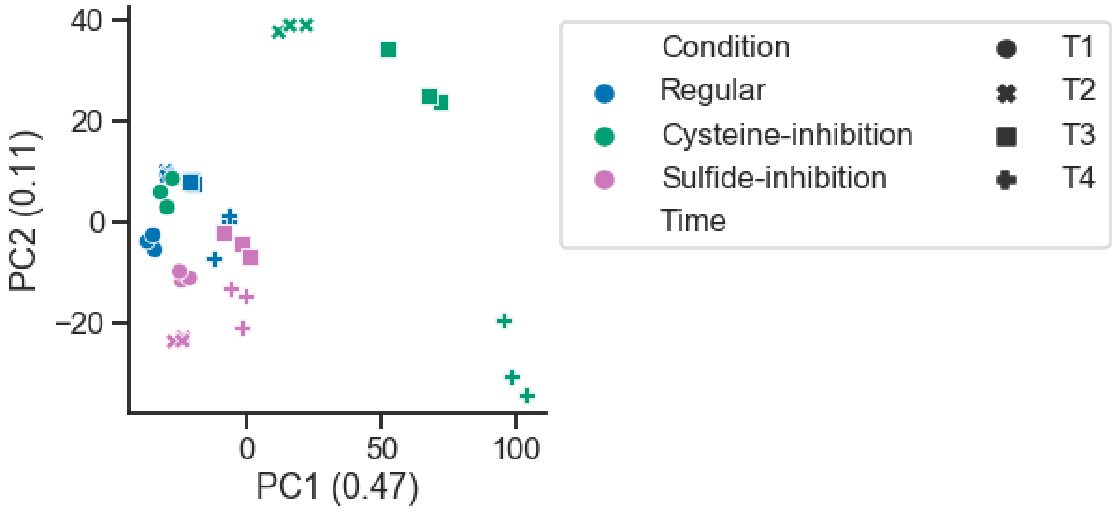
```
Explained Variance… PC1:0.47439388033536134, PC2:0.10675737757105931
```

```
[18]: transcrip_name  Ga0242637_111  Ga0242637_1110  Ga0242637_11100  \
      PC1                  0.002683       -0.003385        -0.013681
      PC2                  0.021117        0.008070         0.000266
```

```
transcrip_name  Ga0242637_111000  Ga0242637_111001  Ga0242637_111002  \
PC1                     0.006148          0.003305          0.016759   
PC2                     0.007217          0.024500          0.020176   

transcrip_name  Ga0242637_111003  Ga0242637_111004  Ga0242637_111005  \
PC1                    -0.020300         -0.001592         -0.022371   
PC2                     0.009323         -0.034581          0.009231   

transcrip_name  Ga0242637_111006  …  Ga0242637_11990  Ga0242637_11991  \
PC1                    -0.022203  …        -0.009629        -0.007276   
PC2                    -0.003576  …        -0.012767        -0.015250   

transcrip_name  Ga0242637_11992  Ga0242637_11993  Ga0242637_11994  \
PC1                    0.019334         0.024265        -0.003648   
PC2                   -0.004965        -0.003579        -0.006621   

transcrip_name  Ga0242637_11995  Ga0242637_11996  Ga0242637_11997  \
PC1                   -0.025158        -0.021452        -0.017951   
PC2                    0.007340         0.012426         0.005154   

transcrip_name  Ga0242637_11998  Ga0242637_11999  
PC1                    0.021064        -0.014212  
PC2                   -0.015337         0.006288  

[2 rows x 3686 columns]
```

[19]:

```
c=sns.color_palette('colorblind')
plt.figure(figsize=(4,4))
sns.
 ↪scatterplot(x='PC1',y='PC2',hue='Condition',style='Time',data=pca_df,palette=[c[0],c[2],c[4
plt.legend(bbox_to_anchor=(1,1),loc=2,ncol=2)
plt.xlabel('PC1 (%s)'%round(pca.explained_variance_ratio_[0],2))
plt.ylabel('PC2 (%s)'%round(pca.explained_variance_ratio_[1],2))
plt.savefig('All-zscore-PCA.pdf',bbox_inches='tight')
```



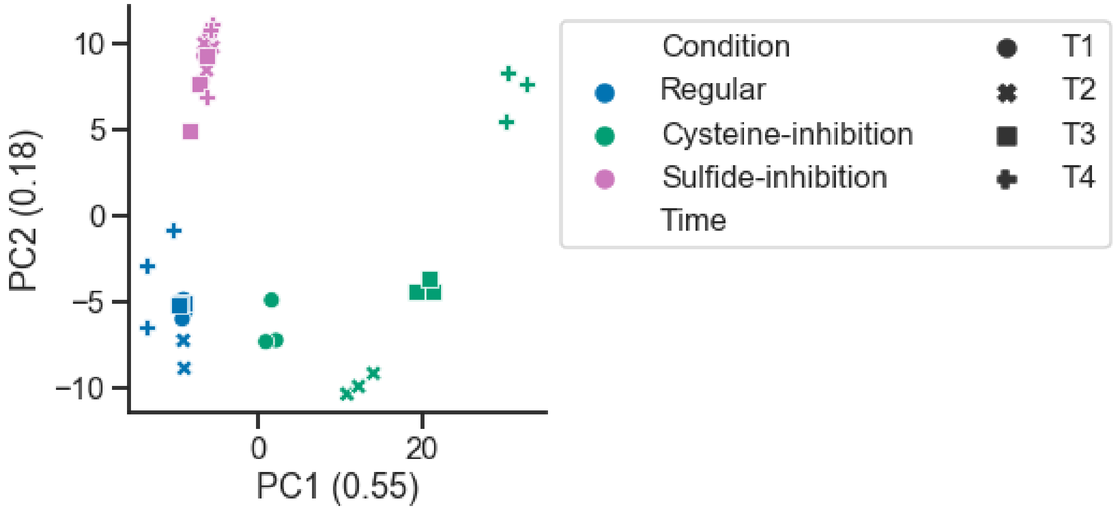
[20]:
```
#Hits identified by Boruta
pca_data =zvals.loc[hits,cond].copy().T
pca = PCA(n_components=2)
pca_df = pd.DataFrame(pca.
 ↪fit_transform(pca_data),columns=['PC1','PC2'],index=pca_data.index)
pca_df=pca_df.reset_index()
pca_df.columns=['Condition','Time','Rep','PC1','PC2']
pca_df['Time']=['T%s'%i for i in pca_df['Time']+1]
print('Explained Variance... PC1:%s, PC2:%s'%tuple(pca.
 ↪explained_variance_ratio_))
pca_weights = pd.DataFrame(pca.components_,index=['PC1','PC2'],columns=pca_data.
 ↪columns)
pca_weights
```

```
Explained Variance… PC1:0.546552642744364, PC2:0.17591062119264678
```

[20]:
```
transcrip_name  Ga0242637_111003  Ga0242637_11105  Ga0242637_111051  \
PC1                    -0.068238        -0.056523         -0.072652
PC2                    -0.036776        -0.051590         -0.020148

transcrip_name  Ga0242637_111053  Ga0242637_111132  Ga0242637_111144  \
PC1                      0.05613         -0.035018         -0.062183
PC2                     -0.04756         -0.032739          0.013036

transcrip_name  Ga0242637_11116  Ga0242637_111188  Ga0242637_111203  \
PC1                    0.070633          0.074716          0.077129
PC2                   -0.003437          0.019979         -0.025816

transcrip_name  Ga0242637_111204  …  Ga0242637_11599  Ga0242637_11625  \
PC1                    0.053594  …        -0.079196        0.046798
PC2                   -0.082269  …         0.014012       -0.038629

transcrip_name  Ga0242637_11666  Ga0242637_11667  Ga0242637_11727  \
PC1                    0.059194         0.057243        -0.047500
PC2                   -0.025023        -0.032512         0.006312

transcrip_name  Ga0242637_11769  Ga0242637_11831  Ga0242637_11878  \
PC1                   -0.025947         0.059607         0.042545
PC2                    0.096434         0.006258        -0.069919

transcrip_name  Ga0242637_11902  Ga0242637_11946
PC1                    0.068775        -0.044623
PC2                   -0.052895         0.035472

[2 rows x 362 columns]
```

[21]:

```
c=sns.color_palette('colorblind')
plt.figure(figsize=(4,4))
sns.
↪scatterplot(x='PC1',y='PC2',hue='Condition',style='Time',data=pca_df,palette=[c[0],c[2],c[4
plt.legend(bbox_to_anchor=(1,1),loc=2,ncol=2)
plt.xlabel('PC1 (%s)'%round(pca.explained_variance_ratio_[0],2))
plt.ylabel('PC2 (%s)'%round(pca.explained_variance_ratio_[1],2))
plt.savefig('Hits-zscore-PCA.pdf',bbox_inches='tight')
```



6 K-means clustering of gene expression data

- Groups genes into k clusters, where k is user defined

- Like above methods, can be useful for reducing dimensions of data
- We are using it to further reduce dimensionality of hits identified by Boruta
- Clusters should contain genes with similair expression patterns across condition
- With groups of genes that behave roughly the same we can more easily look at trends and figure out if the group is associated with certain pathways or functions
- For clustering below expression values for each gene across all conditions are appended to create one long vector of values (expression of each gene across all conditions). This allows the algorithm to find genes that behave similarly across conditions

[27]:

```
def␣
 ↪cluster_genes(df,cols=['Regular','Cysteine-inhibition','Sulfide-inhibition'],k=8):
 ↪
    X = df.values
    km = KMeans(n_clusters=k, random_state=0)
    return km.fit_predict(X)
```

[644]:

```
#Select colors for plotting
cond_labels = [x[0] for x in zvals.columns.values]
cond=['Regular','Cysteine-inhibition','Sulfide-inhibition']
colors = sns.color_palette('colorblind')
cdict = {x:colors[i] for i,x in enumerate(pd.unique(cond_labels))}
#Parameters for creating plot grid
scale=5
figcols = 4
clust_res = []
k_vals = []
inertias = []
df = zvals.loc[hits,:].copy()
df=df[cond]
#Loop through and try clustersing genes into 2 to 40 clusters and display␣
 ↪results
plot=False #Set equal to True to visualize the clustering
for n in progbar(range(2,40)):
    print('n=%s'%n)
    X = df.values
    km = KMeans(n_clusters=n, random_state=0)
    y_pred=km.fit_predict(X)
    clust_res.append(y_pred)
    #Create grid of plots
    if plot:
        figrows = int(round((n+1)/figcols,0))
        if (n+1)/figcols -figrows >0:
            figrows+=1
        fig = plt.figure(figsize=(figcols*scale,figrows*scale))
        #Loop through grid and plot clustering results

        for i in range(n):
            ax = plt.subplot(figrows,figcols,i+1)
            clust=X[y_pred==i]
            #labels =np.array(cond_labels)[y_pred==i]
            #label_counts= {x:0 for i,x in enumerate(pd.unique(cond_labels))}
            #count_sum = 0
            #Only plot 100 lines to speed things up, if more then 100 present␣
↪select random set
            if len(clust)<=100:
                itr = range(len(clust))
            else:
                itr = np.random.choice(range(0,len(clust)),100)
            for x in clust[itr,:]:
                ax.plot(x,alpha=0.3)
            #Plot average behavior for cluster
            ax.plot(km.cluster_centers_[i].ravel(), "k--")
            ax.set_title('cluster %s, n=%s'%(i+1,len(clust)))
        fig.tight_layout()
        #plt.savefig('TS-Kmeans-Clustering-n-%s-euclid.
↪pdf'%n,dpi=1000,bbox_inches='tight')
        #show and then clear figure to speed things up
        plt.show()
        plt.clf()
    #Keep track of k value and clustering score (intertia)
    k_vals.append(n)
    inertias.append(km.inertia_)
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:16:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  app.launch_new_instance()
```

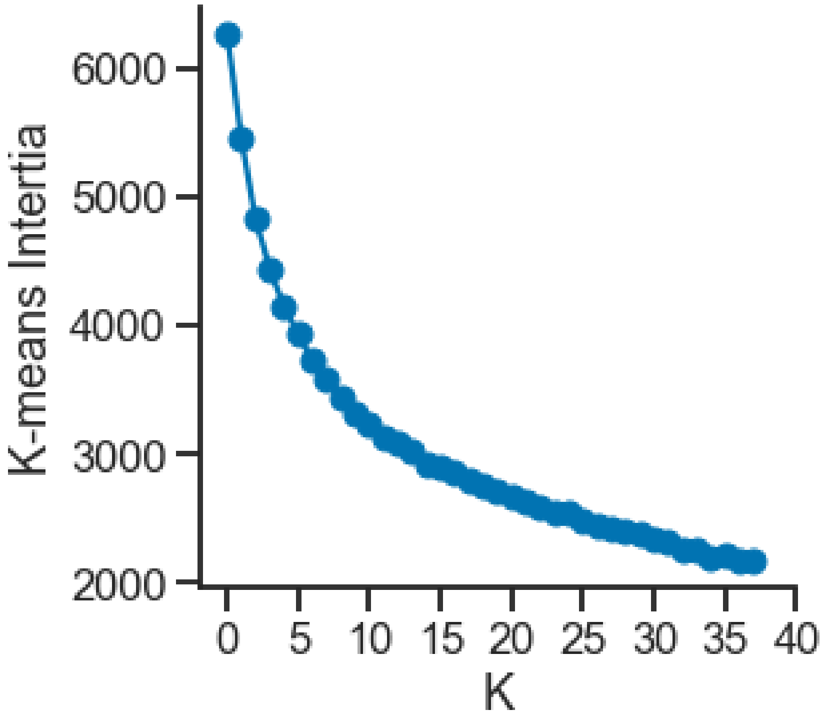
```
HBox(children=(FloatProgress(value=0.0, max=38.0), HTML(value='')))
```

```
n=2
n=3
n=4
n=5
n=6
n=7
n=8
n=9
n=10
n=11
n=12
n=13
n=14
n=15
```

```
n=16
n=17
n=18
n=19
n=20
n=21
n=22
n=23
n=24
n=25
n=26
n=27
n=28
n=29
n=30
n=31
n=32
n=33
n=34
n=35
n=36
n=37
n=38
n=39
```

[645]:

```
#Clustering results
#Results suggest there is not a well defined value for K
plt.figure(figsize=(4,4))
plt.plot(range(len(inertias)),inertias,marker='o')
plt.ylabel('K-means Intertia')
plt.xlabel('K')
xticks= plt.xticks([0,5,10,15,20,25,30,35,40])
```



7 Visualizing clustering results with a heatmap

- While not ideal, based on above results, k=5 was choosen as it captures many of the major patterns in the data and each cluster still contains a large number of genes for subsequent enrichment analyis
- Can change number of clusters below by changing k

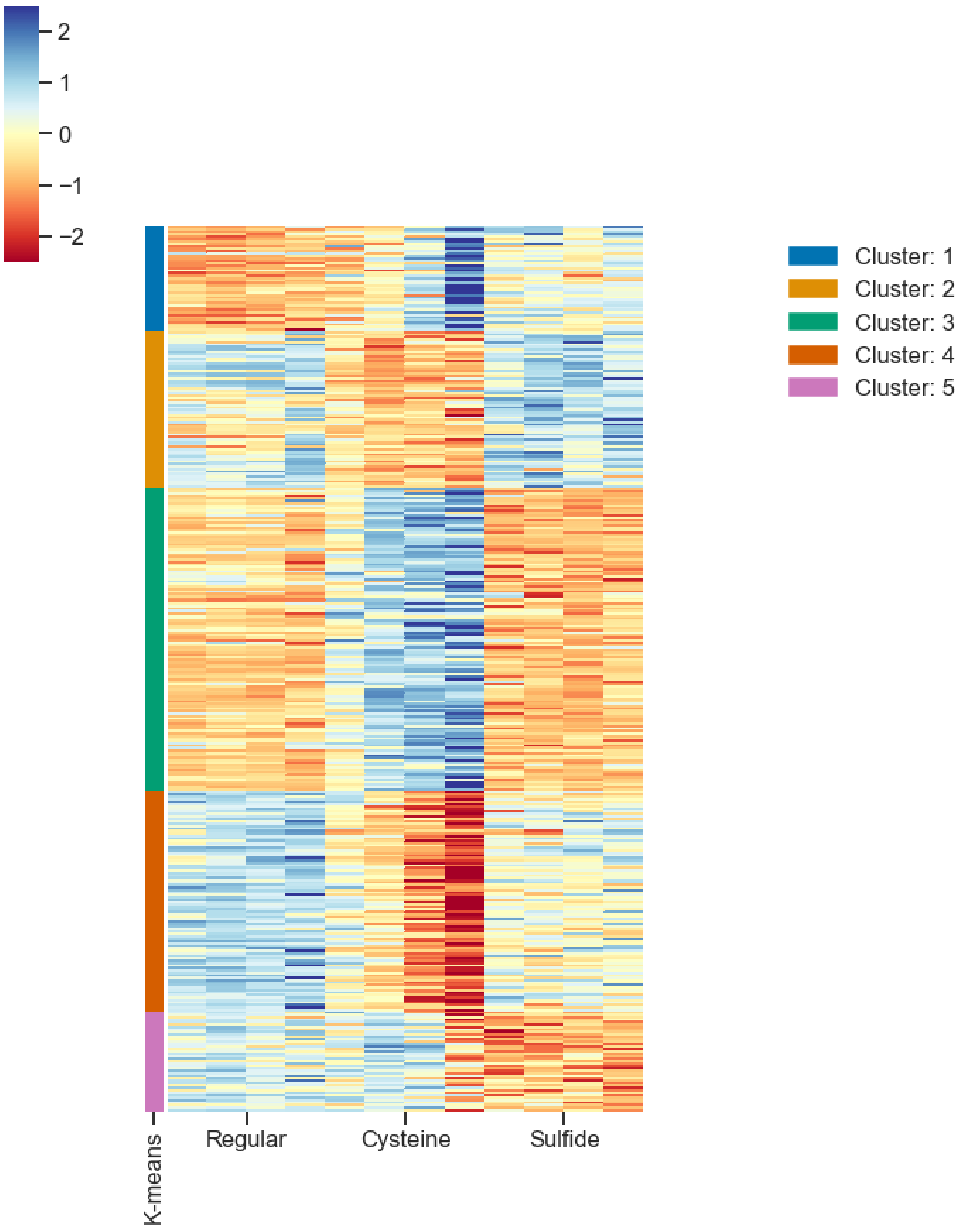
[277]:

```
colors = sns.color_palette('colorblind')
colors[5]=colors[7]
cond=['Regular','Cysteine-inhibition','Sulfide-inhibition']
#Select data table to use and format for clustering
k=5
#cond=['Regular','Sulfide-inhibition','Sulfide-injection']
y_pred = cluster_genes(zvals.loc[hits,cond],k=k)
df=avgzvals.loc[hits,:]
df=df[cond]
#map colors to cluster number, if k > len(colors) then will have an issue. Can use plt.cm.tab20 for k>10
#If using plt.cm.tab20 then colors[i] must be changed to colors(i)
cmap = {df.index[i]:colors[y_pred[i]] for i in range(len(df.index))}
handles = [mpatches.Patch(color=colors[i], label='Cluster: %s'%(i+1)) for i in range(k)]
df['kmeans'] =y_pred
```

```
df = df.sort_values('kmeans').drop('kmeans',axis=1)
row_colors = pd.Series(df.index).map(cmap)
row_colors.index=df.index
row_colors.name='K-means'
#Creat heatmap, grouping by cluster which is colored in furthers left column
clust = sns.
 ↪clustermap(df,cmap='RdYlBu',yticklabels='',figsize=(7,14),row_colors=row_colors,col_cluster
 ↪5,vmin=2.5)
clust.ax_heatmap.set(ylabel='',xlabel='',xticks=[2,6,10])
clust.ax_heatmap.set_xticklabels([x.split('-')[0] for x in cond],rotation=0)
clust.ax_heatmap.legend(bbox_to_anchor=(1.7,1),handles=handles,frameon=False)
#plt.savefig('070920-Kmeans-Heatmap-K%s.pdf'%(k),bbox_inches='tight')
```

```
//anaconda3/lib/python3.7/site-packages/pandas/core/generic.py:3946:
PerformanceWarning: dropping on a non-lexsorted multi-index without a level
parameter may impact performance.
  new_axis = axis.drop(labels, errors=errors)
```

```
[277]: <matplotlib.legend.Legend at 0x1a22d11198>
```


2
1
0
−1
−2
Cluster: 1
Cluster: 2
Cluster: 3
Cluster: 4
Cluster: 5
K-means
Regular
Cysteine
Sulfide

8 Enrichment Analysis

```
[636]: #k-means cluster term enrichment using hypergeometric test
       #Returns results for terms enriched in each cluster, along with avg zscores,␣
        ↪and FDR corrected enrichment p-value
       def Enrichment(terms,avgzvals,hits,y_pred,p=0.05,enrich='GO Biological␣
        ↪Terms',seed=False,level=None):
           k=len(set(y_pred))
           clustdf = pd.DataFrame()
           pvals = []
           for clust in progbar(range(1,k+1)):
               #For each cluster get the associated genes
               genes =  avgzvals.loc[hits,:].index[y_pred==clust-1]

               #M is the population size
               #n is the number of successes in the population
               #N is the sample size
               #X is the number of drawn "successes".
               idx = []

               #For each gene figure out what its position(s) is(are) in the␣
        ↪annotation table
               for g in genes:
                   ix = list(terms[terms['ID'] == g].index.values)
                   idx+=ix

               #Count how many times each term shows up in table for the set of genes
               pathcounts =terms.loc[idx,:].groupby(by='value').count()

               #Get sample and population size params
               N=len(pathcounts)
               M=len(terms['value'])
               IDs = []
               smp_count =[]
               null_count =[]
               pvals=[]

               #Loop  through each term and test if it's significantly enriched using␣
        ↪the hypergeometric test
               for val in pathcounts.index:
                   n =terms[terms['value']==val].dropna().shape[0]

                   #Don't test terms with less then 3 representative, could be␣
        ↪problematic
                   if val in IDs or n <3:
                       continue
                   IDs.append(val)
```

```
            x = pathcounts.loc[val,'ID']
            smp_count.append(x)
            null_count.append(n)
            pval = hypergeom.sf(x-1, M, n, N)
            pvals.append(pval)

        #Correct p-values
        smp_count=np.array(smp_count)
        null_count=np.array(null_count)
        IDs = np.array(IDs)
        df = pd.DataFrame(np.
↪array([IDs,pvals,len(pvals)*[clust],smp_count,null_count]).
↪T,columns=['name','pval','clust','clust_n','null_n'])
        df['logp'] = -np.log(df['pval'].astype(float))
        clustdf = pd.concat([clustdf,df],ignore_index=True)

    #Format clustdf values and calculate percentage of genes in each category␣
↪found in the cluster
    clustdf['clust_n']=clustdf['clust_n'].astype(float)
    clustdf['null_n']=clustdf['null_n'].astype(float)
    clustdf['percent'] = clustdf['clust_n']/clustdf['null_n']

    #Sort values and add logp
    clustdf=clustdf.sort_values(by='pval')
    rej, pval_corr = multipletests(clustdf['pval'].values.astype(float),␣
↪method='fdr_bh',alpha=0.01)[:2]
    clustdf['pval']=pval_corr
    clustdf=clustdf[clustdf['pval']<=p]
    clustdf['logp']=np.log(clustdf['pval'].values)

    #Print results
    for clust,df in clustdf.groupby(by='clust'):
        genes =  avgzvals.loc[hits,:].index[y_pred==int(clust)-1]
        print('Cluster %s: %s\n'%(int(clust),len(genes)))
        for i in df.index.values:
            print(df.loc[i,'name'],df.loc[i,'pval'],df.loc[i,'clust_n'])
        print('\n--------------------------------------------------------')

    #Add avg z-scores and genes
    clustdf['avg_zscore_cys']=np.nan
    clustdf['avg_zscore_sulf']=np.nan
    clustdf['avg_zscore_nt']=np.nan
    clustdf['genes']=np.nan
    clustdf['enrichement']=enrich
    for i in clustdf.index:
        name=clustdf.loc[i,'name']
        g=list(set(terms[terms['value']==name].dropna().ID.unique())&set(hits))
        g_new=[gene_map[x] for x in g]
        clustdf.loc[i,'genes']=';'.join(g_new)
        vals=avgzvals.loc[genes,cond].groupby(level=[0],axis=1).mean().mean().
↪reset_index()
        clustdf.
↪loc[i,['avg_zscore_cys','avg_zscore_sulf','avg_zscore_nt']]=vals[0].values
    return clustdf
```

9 Enrichment using GO biological terms

[597]:

```
print('Go Biological Terms')
terms=[]
for g in sets[('Regular', 'Cysteine-inhibition')]:
    vals=list(biol[biol.ID==g]['value'].unique())
    terms+=vals
print('Cysteine: %s'%(len(pd.unique(terms))))

terms=[]
for g in sets[('Regular', 'Sulfide-inhibition')]:
    vals=list(biol[biol.ID==g]['value'].unique())
    terms+=vals
print('Sulfide: %s'%(len(pd.unique(terms))))

print('Total unique terms: %s'%(len(biol['value'].unique())))
```

```
Go Biological Terms
Cysteine: 117
Sulfide: 85
Total unique terms: 545
```

[637]:

```
#Go biological terms
clustdf=Enrichment(biol,avgzvals,hits,y_pred)

#Add results to supplementary table
supp_table=pd.DataFrame() #initalize
supp_table=pd.concat([supp_table,clustdf],ignore_index=True)

#Visualize significant hits using fraction of genes in each category found in␣
↪the cluster
for index, df in clustdf.groupby(by='clust'):
    df['sort']=[x.split()[-1] for x in df['name']]
    df=df.sort_values(by=['sort','pval'])
    plt.figure(figsize=(4,.5*len(df)))
    g = sns.
↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
    plt.ylabel('')
    plt.xlabel('Fraction of Annotated Genes')
    plt.xlim(0,1)
    #plt.savefig('Pathway-Clust%s-K-%s.pdf'%(index,k),bbox_inches='tight')
```

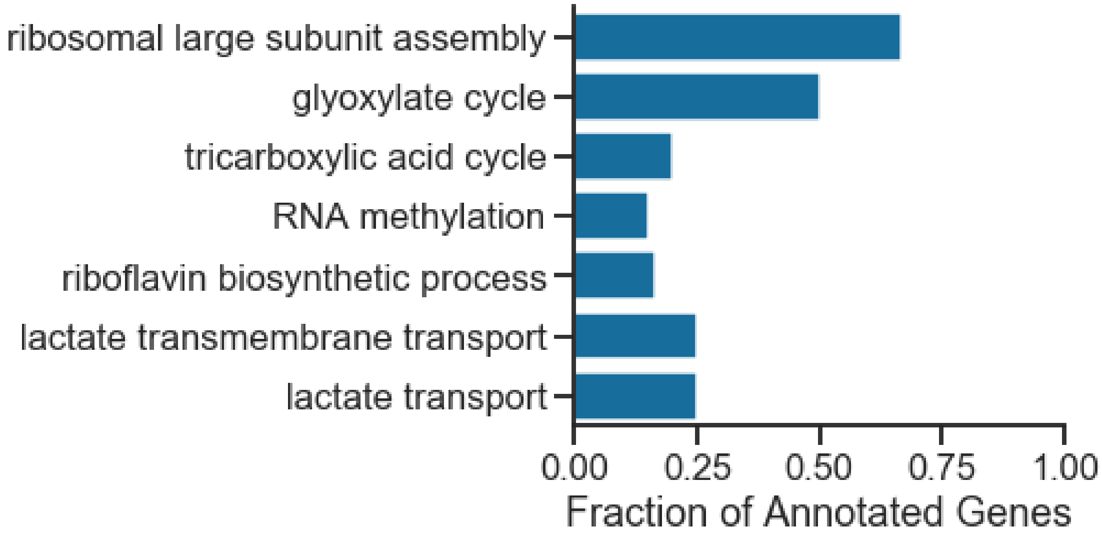
```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  import sys
```

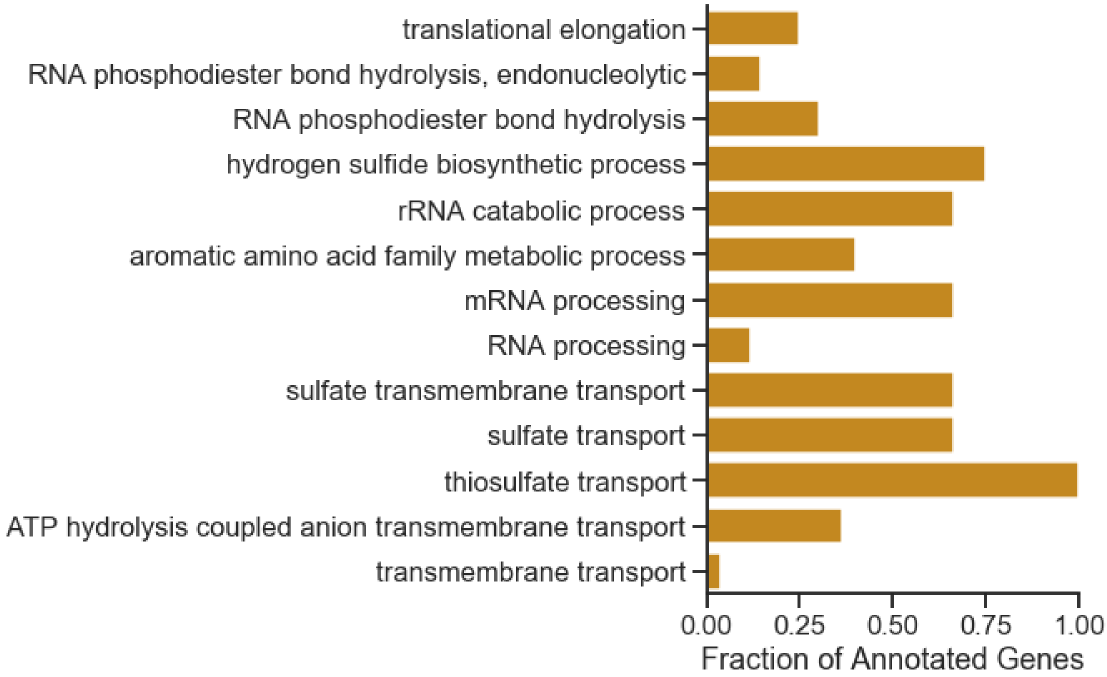
```
HBox(children=(FloatProgress(value=0.0, max=5.0), HTML(value='')))
```

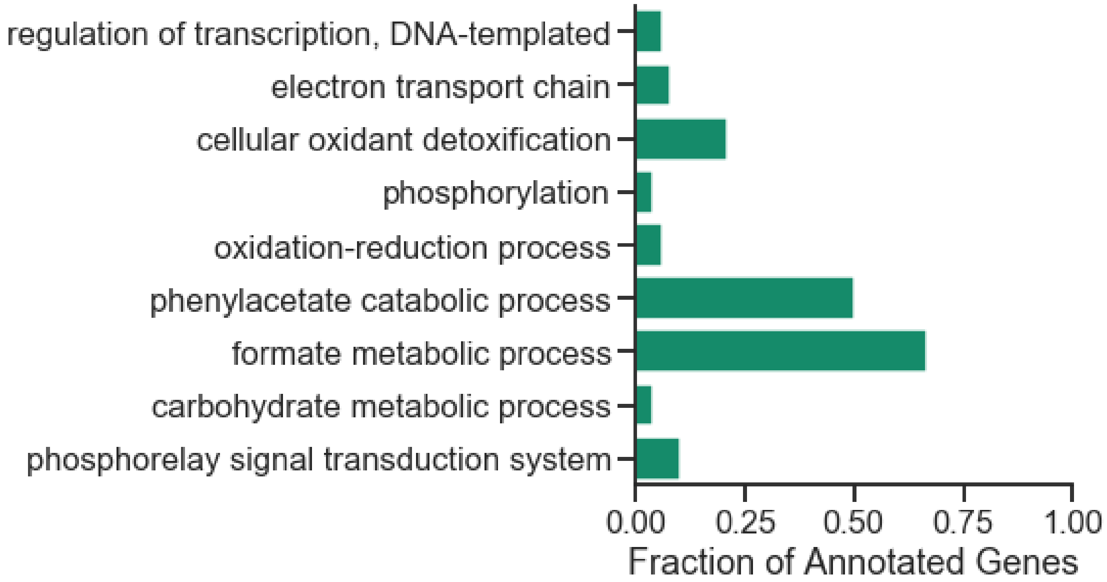
```


Cluster 1: 43

ribosomal large subunit assembly 0.0014185652783221556 2.0
riboflavin biosynthetic process 0.01750919368240511 2.0
RNA methylation 0.019969821035716425 2.0
glyoxylate cycle 9.659127101892555e-05 3.0
lactate transmembrane transport 0.000647606655644228 3.0
lactate transport 0.000647606655644228 3.0
tricarboxylic acid cycle 0.0012448738615162104 3.0

--------------------------------------------------------
Cluster 2: 64

RNA phosphodiester bond hydrolysis 0.0019472240340117455 3.0
mRNA processing 0.003969480640379756 2.0
rRNA catabolic process 0.003969480640379756 2.0
transmembrane transport 0.006098700987398663 10.0
aromatic amino acid family metabolic process 0.011171452355338505 2.0
RNA phosphodiester bond hydrolysis, endonucleolytic 0.013782157904895651 3.0
RNA processing 0.020958196609023063 3.0
translational elongation 0.024464949925397498 2.0
thiosulfate transport 4.516867194518953e-05 3.0
sulfate transmembrane transport 9.88776344272913e-06 4.0
sulfate transport 9.88776344272913e-06 4.0
ATP hydrolysis coupled anion transmembrane transport 0.0001135341964958861 4.0
hydrogen sulfide biosynthetic process 0.00011944340336934562 3.0

--------------------------------------------------------
Cluster 3: 124

formate metabolic process 0.001860970421910687 2.0
phosphorylation 0.0036625255305099396 7.0
electron transport chain 0.004597523974068674 4.0
carbohydrate metabolic process 0.037898099625193686 4.0
oxidation-reduction process 2.4843686529757275e-22 32.0
```

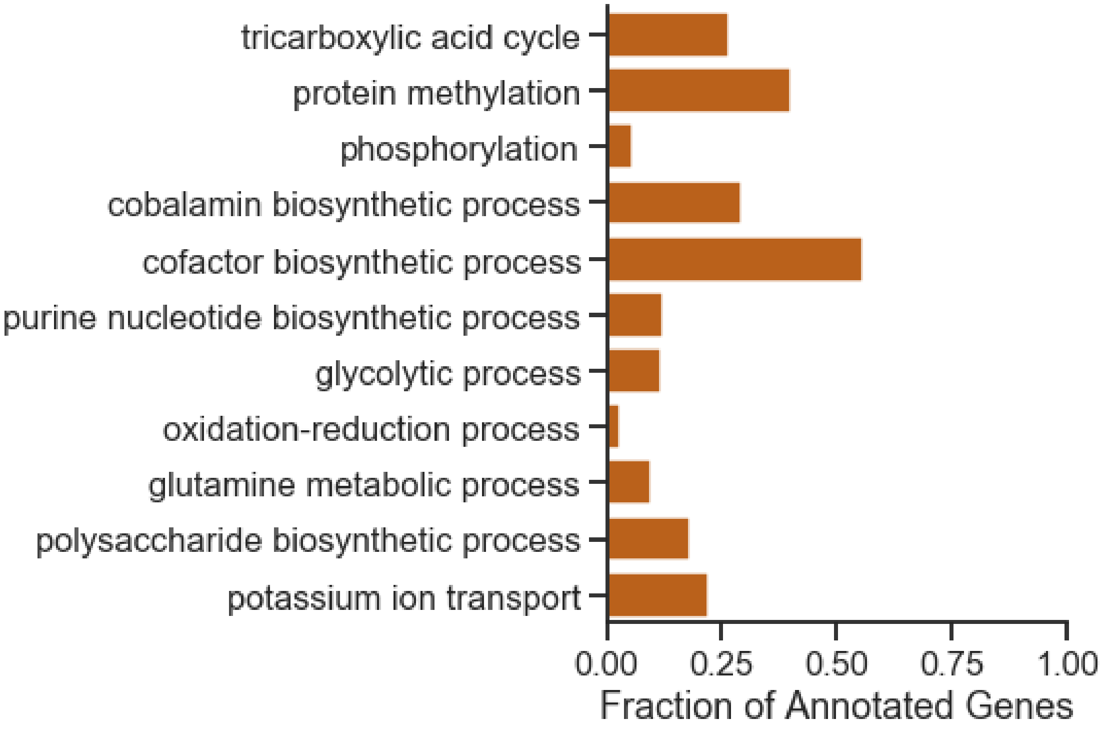
```
regulation of transcription, DNA-templated 1.6489313299896614e-09 16.0
phosphorelay signal transduction system 8.652636836463813e-05 6.0
cellular oxidant detoxification 2.152880732594035e-05 5.0
phenylacetate catabolic process 0.00013146139175537912 3.0

--------------------------------------------------------
Cluster 4: 90

phosphorylation 0.001451340729506865 10.0
protein methylation 0.013194322753386263 2.0
purine nucleotide biosynthetic process 0.024087606963936165 3.0
glycolytic process 0.025614212289295453 3.0
potassium ion transport 0.036005979505386976 2.0
oxidation-reduction process 0.037898099625193686 14.0
glutamine metabolic process 0.04159194219166808 3.0
polysaccharide biosynthetic process 0.04765982867528787 2.0
cobalamin biosynthetic process 1.0359855665867165e-06 7.0
tricarboxylic acid cycle 0.0005326768286505848 4.0
cofactor biosynthetic process 2.250888609789538e-06 5.0

--------------------------------------------------------
Cluster 5: 41

glycerol metabolic process 0.008724627371840285 2.0
transmembrane transport 0.04048276298321134 5.0

--------------------------------------------------------

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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
```

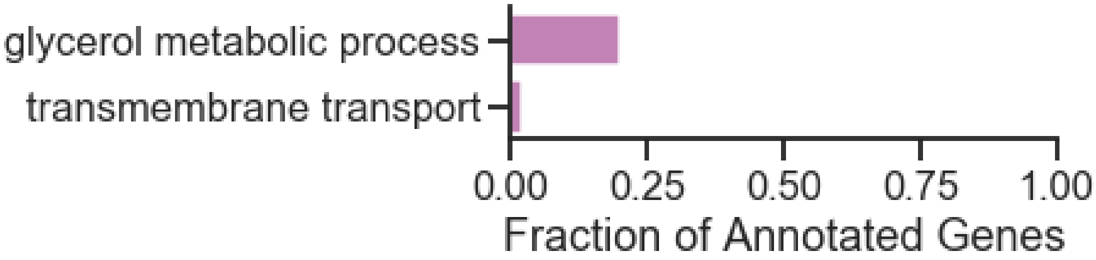
```
See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
SettingWithCopyWarning:
A value is trying 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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```




translational elongation
RNA phosphodiester bond hydrolysis, endonucleolytic
RNA phosphodiester bond hydrolysis
hydrogen sulfide biosynthetic process
rRNA catabolic process
aromatic amino acid family metabolic process
mRNA processing
RNA processing
sulfate transmembrane transport
sulfate transport
thiosulfate transport
ATP hydrolysis coupled anion transmembrane transport
transmembrane transport
0.00
0.25
0.50
0.75
1.00
Fraction of Annotated Genes


regulation of transcription, DNA-templated
electron transport chain
cellular oxidant detoxification
phosphorylation
oxidation-reduction process
phenylacetate catabolic process
formate metabolic process
carbohydrate metabolic process
phosphorelay signal transduction system
0.00
0.25
0.50
0.75
1.00
Fraction of Annotated Genes





10 Enrichment using MetaCyc terms

[630]:
```
print('MetaCyc Pathway and Parent-pathway Terms')
terms=[]
for g in sets[('Regular', 'Cysteine-inhibition')]:
    vals=list(path_annot[path_annot.ID==g]['value'].unique())
    terms+=vals
print('Cysteine: %s'%(len(pd.unique(terms))))

terms=[]
for g in sets[('Regular', 'Sulfide-inhibition')]:
    vals=list(path_annot[path_annot.ID==g]['value'].unique())
    terms+=vals
print('Sulfide: %s'%(len(pd.unique(terms))))

print('Total unique terms: %s'%(len(path_annot['value'].unique())))
```

```
MetaCyc Pathway and Parent-pathway Terms
Cysteine: 155
Sulfide: 79
Total unique terms: 477
```

[882]:

```
#MetaCyc pathway terms
pathways = path_annot[path_annot['type']=='pathway']
clustdf=Enrichment(pathways,avgzvals,hits,y_pred,enrich='MetaCyc Pathway')

#Add results to supplementary table
supp_table=pd.concat([supp_table,clustdf],ignore_index=True)

#Visualize significant hits using fraction of genes in each category found in␣
 ↪the cluster
for index, df in clustdf.groupby(by='clust'):
    df['sort']=[x.split()[-1] for x in df['name']]
    df=df.sort_values(by=['sort','pval'])
    plt.figure(figsize=(4,.5*len(df)))
    g = sns.
 ↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
    plt.ylabel('')
    plt.xlabel('Fraction of Annotated Genes')
    plt.xlim(0,1)
    #plt.savefig('Pathway-Clust%s-K-%s.pdf'%(index,k),bbox_inches='tight')
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  import sys
```

```
HBox(children=(FloatProgress(value=0.0, max=39.0), HTML(value='')))
```

```
//anaconda3/lib/python3.7/site-packages/pandas/core/series.py:856:
RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

```
Cluster 1: 20

nitrate reduction X (dissimilatory, periplasmic) 0.0015737772076696026 3.0
nitrate reduction V (assimilatory) 0.0016390199969527897 3.0
phenylacetate degradation I (aerobic) 0.04231651327208994 2.0
```

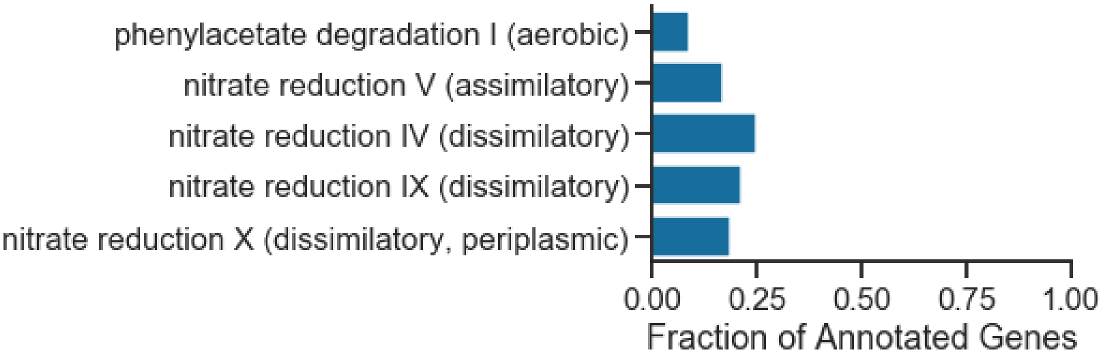
```
nitrate reduction IV (dissimilatory) 0.000880588276630345 3.0
nitrate reduction IX (dissimilatory) 0.0011610651977693664 3.0

--------------------------------------------------------
Cluster 11: 8

L-lysine biosynthesis I 0.024695592522706113 1.0

--------------------------------------------------------
Cluster 12: 15

glycolysis I (from glucose 6-phosphate) 0.04974979907847867 2.0
TCA cycle V (2-oxoglutarate:ferredoxin oxidoreductase) 0.04974979907847867 2.0

--------------------------------------------------------
Cluster 13: 13

formate oxidation to CO<sub>2</sub> 0.006540994776284326 1.0

--------------------------------------------------------
Cluster 15: 3

adenosine deoxyribonucleotides <i>de novo</i> biosynthesis 0.04094537232042565
1.0
guanosine deoxyribonucleotides <i>de novo</i> biosynthesis I 0.04094537232042565
1.0

--------------------------------------------------------
Cluster 18: 11

glycolysis III (from glucose) 0.038655154055455575 2.0

--------------------------------------------------------
Cluster 2: 12

adenosylcobinamide-GDP salvage from cobinamide I 0.00039537346113221317 3.0
adenosylcobinamide-GDP biosynthesis from cobyrinate <i>a,c</i>-diamide
0.0008344003937487271 3.0
adeninyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.00018268509339667787 3.0
5-methoxy-6-methylbenzimidazolyl adenosylcobamide biosynthesis from
adenosylcobinamide-GDP 0.00018268509339667787 3.0
5-methoxybenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-
GDP 0.00018268509339667787 3.0
5-methylbenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.00018268509339667787 3.0
benzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.00018268509339667787 3.0
```

```
2-methyladeninyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.00018268509339667787 3.0
5-hydroxybenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-
GDP 0.00018268509339667787 3.0
adenosylcobalamin biosynthesis from adenosylcobinamide-GDP I
0.00018268509339667787 3.0

-------------------------------------------------------
Cluster 20: 13

flavin biosynthesis I (bacteria and plants) 0.01815126050418903 1.0

-------------------------------------------------------
Cluster 22: 7

L-homoserine biosynthesis 0.04231651327208994 1.0

-------------------------------------------------------
Cluster 23: 10

ppGpp biosynthesis 0.01815126050418903 1.0

-------------------------------------------------------
Cluster 24: 8

TCA cycle V (2-oxoglutarate:ferredoxin oxidoreductase) 0.0 2.0

-------------------------------------------------------
Cluster 26: 9

formate oxidation to CO<sub>2</sub> 0.02198303298819057 1.0
L-alanine degradation IV 0.02198303298819057 1.0
preQ<sub>0</sub> biosynthesis 0.0008344003937487271 2.0

-------------------------------------------------------
Cluster 28: 2

methylerythritol phosphate pathway I 0.043644108504439894 1.0
methylerythritol phosphate pathway II 0.043644108504439894 1.0

-------------------------------------------------------
Cluster 30: 8

pseudouridine degradation 0.02198303298819057 1.0
D-xylose degradation I 0.024695592522706113 1.0

-------------------------------------------------------
Cluster 32: 3
```

```
dTDP-L-rhamnose biosynthesis 0.010614772224671913 1.0

--------------------------------------------------------
Cluster 35: 10

glycerol-3-phosphate to cytochrome <i>bo</i> oxidase electron transfer
0.0019384674056728553 2.0
nitrate reduction IX (dissimilatory) 0.0030084653920353114 2.0
nitrate reduction X (dissimilatory, periplasmic) 0.003845209239910862 2.0
glycerol degradation I 0.0008344003937487271 2.0
glycerophosphodiester degradation 0.0010007504369470956 2.0
glycerol-3-phosphate to fumarate electron transfer 0.0010007504369470956 2.0
glycerol and glycerophosphodiester degradation 0.0011895571912629642 2.0

--------------------------------------------------------
Cluster 36: 26

phenylacetate degradation I (aerobic) 0.03568196509370484 1.0

--------------------------------------------------------
Cluster 37: 5

5-methoxy-6-methylbenzimidazolyl adenosylcobamide biosynthesis from
adenosylcobinamide-GDP 0.0016390199969527897 2.0
adeninyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.0016390199969527897 2.0
adenosylcobalamin biosynthesis from adenosylcobinamide-GDP I
0.0016390199969527897 2.0
5-methylbenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.0016390199969527897 2.0
benzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.0016390199969527897 2.0
2-methyladeninyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
0.0016390199969527897 2.0
5-hydroxybenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-
GDP 0.0016390199969527897 2.0
5-methoxybenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-
GDP 0.0016390199969527897 2.0
adenosylcobinamide-GDP salvage from cobinamide I 0.0026299978860069564 2.0
adenosylcobinamide-GDP biosynthesis from cobyrinate <i>a,c</i>-diamide
0.0043961084856769935 2.0

--------------------------------------------------------
Cluster 4: 7

flavin biosynthesis I (bacteria and plants) 0.02900295558713828 1.0
```

```
--------------------------------------------------------
Cluster 5: 7

glycerophosphodiester degradation 0.02198303298819057 1.0
glycerol and glycerophosphodiester degradation 0.024527662589861325 1.0

--------------------------------------------------------
Cluster 6: 20

glycolysis I (from glucose 6-phosphate) 0.024527662589861325 2.0
glycolysis III (from glucose) 0.024695592522706113 2.0
superpathway of glucose and xylose degradation 0.029414646588461716 2.0

--------------------------------------------------------
Cluster 7: 14

adenosylcobalamin salvage from cobalamin 0.04743396539403594 1.0

--------------------------------------------------------
Cluster 8: 4

TCA cycle V (2-oxoglutarate:ferredoxin oxidoreductase) 0.0016390199969527897 2.0
mixed acid fermentation 0.0011610651977693664 2.0

--------------------------------------------------------
Cluster 9: 12

selenate reduction 0.04422070628775117 1.0
assimilatory sulfate reduction III 4.4981937983957936e-05 3.0

--------------------------------------------------------

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:65:
RuntimeWarning: divide by zero encountered in log
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
```

```
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.


       ␣
 ↪---------------------------------------------------------------------------


        IndexError                                Traceback (most recent call␣
 ↪last)


        <ipython-input-882-0b7ffcd1b563> in <module>
         11     df=df.sort_values(by=['sort','pval'])
         12     plt.figure(figsize=(4,.5*len(df)))
    ---> 13     g = sns.
 ↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
         14     plt.ylabel('')
         15     plt.xlabel('Fraction of Annotated Genes')


        IndexError: list index out of range
```



```
<Figure size 288x36 with 0 Axes>
```

[889]:

```
#MetaCyc parent-pathway terms
pathways = path_annot[path_annot['type']=='parent-pathway']
clustdf=Enrichment(pathways,avgzvals,hits,y_pred,enrich='MetaCyc Parent␣
 ↪Pathway')

#Add results to supplementary table
supp_table=pd.concat([supp_table,clustdf],ignore_index=True)

colors=sns.color_palette('colorblind',)
```

```
#Visualize significant hits using fraction of genes in each category found in␣
 ↪the cluster
for index, df in clustdf.groupby(by='clust'):
    df['sort']=[x.split()[-1] for x in df['name']]
    df=df.sort_values(by=['sort','pval'])
    plt.figure(figsize=(4,.5*len(df)))
    g = sns.
 ↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
    plt.ylabel('')
    plt.xlabel('Fraction of Annotated Genes')
    plt.xlim(0,1)
    #plt.savefig('Pathway-Clust%s-K-%s.pdf'%(index,k),bbox_inches='tight')
```

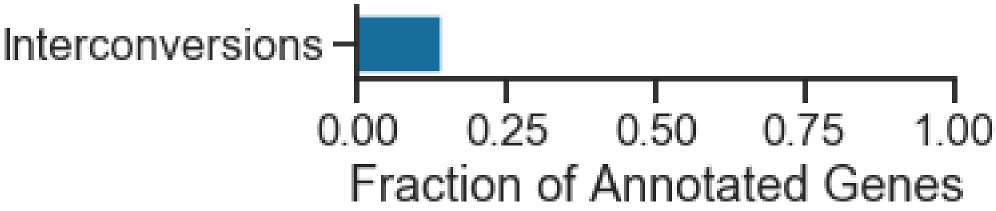
```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  import sys

HBox(children=(FloatProgress(value=0.0, max=39.0), HTML(value='')))


//anaconda3/lib/python3.7/site-packages/pandas/core/series.py:856:
RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:65:
RuntimeWarning: divide by zero encountered in log


Cluster 1: 20

Interconversions 0.04692880863335496 1.0

--------------------------------------------------------
Cluster 10: 5

L-lysine Degradation 0.01946028107158969 2.0
Fermentation to Butanoate 0.01946028107158969 2.0
Fermentation to Acetate 0.019983936389439436 2.0
Fatty Acid Degradation 0.02486992754981086 2.0

--------------------------------------------------------
Cluster 12: 15

Inosine-5'-phosphate Biosynthesis 0.013439952627297393 2.0
TCA cycle 0.019983936389439436 2.0
Glycolysis 0.03851448165920338 2.0

--------------------------------------------------------
Cluster 14: 7
```

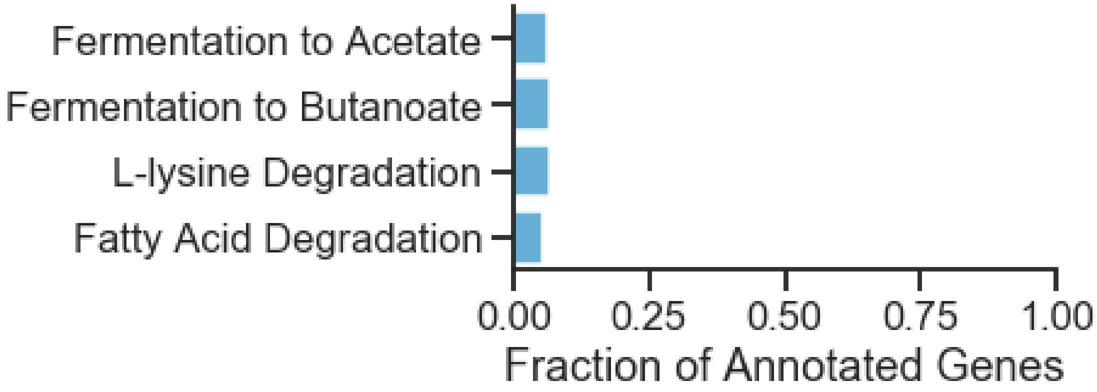
```
L-threonine Degradation 0.029695306658909605 1.0
cob(II)yrinate <i>a,c</i>-diamide biosynthesis 0.04548367846087289 1.0

--------------------------------------------------------
Cluster 15: 3

Pyrimidine Deoxyribonucleotide <i>De Novo</i> Biosynthesis 0.010602753145662433
2.0
Superpathways 0.021347215588118494 4.0
Purine Nucleotide <i>De Novo</i> Biosynthesis 0.0009392787715701704 3.0

--------------------------------------------------------
Cluster 16: 8

L-glutamate Degradation 0.043858025513584656 1.0

--------------------------------------------------------
Cluster 18: 11

Glycolysis 0.04252245270965637 2.0

--------------------------------------------------------
Cluster 2: 12

Adenosylcobamide Biosynthesis 4.0815689005085004e-13 8.0

--------------------------------------------------------
Cluster 23: 10

Metabolic Regulator Biosynthesis 0.012109795479022925 1.0

--------------------------------------------------------
Cluster 25: 2

Ammonia Assimilation 0.021347215588118494 1.0
L-glutamine Biosynthesis 0.021347215588118494 1.0

--------------------------------------------------------
Cluster 26: 9

C1 Compound Utilization and Assimilation 0.04465818693720167 1.0

--------------------------------------------------------
Cluster 27: 9

Glycolysis 0.0 2.0
```

```
--------------------------------------------------------
Cluster 28: 2

Methylerythritol Phosphate Pathways 0.0 2.0

--------------------------------------------------------
Cluster 29: 15

cob(II)yrinate <i>a,c</i>-diamide biosynthesis 0.013439952627297393 2.0
4-Aminobutanoate Degradation 8.81788931214831e-07 4.0

--------------------------------------------------------
Cluster 31: 11

Carboxylate Degradation 0.01654788849762285 2.0

--------------------------------------------------------
Cluster 32: 3

dTDP-sugar Biosynthesis 0.01404034258437439 1.0

--------------------------------------------------------
Cluster 34: 8

Nitrate Reduction 0.0 4.0
Electron Transfer Chains 0.03558672601467782 2.0
Anaerobic Respiration 3.091929657336202e-05 3.0

--------------------------------------------------------
Cluster 35: 10

Electron Transfer Chains 0.002809971767291062 4.0
Nitrate Reduction 0.019983936389439436 2.0
Fatty Acid and Lipid Degradation 0.040235678833975067 1.0
Anaerobic Respiration 0.0006502433128816019 3.0
Glycerol Degradation 0.00011850060328734045 3.0

--------------------------------------------------------
Cluster 39: 6

thiamine Diphosphate Salvage 0.026148092058856685 1.0
Thiamine Diphosphate Biosynthesis 0.0006502433128816019 2.0

--------------------------------------------------------
Cluster 4: 7

Flavin Biosynthesis 0.026148092058856685 1.0
TCA cycle 0.04692880863335496 1.0
```

```
--------------------------------------------------------
Cluster 5: 7

Glycerol Degradation 0.0005738621444024156 2.0

--------------------------------------------------------
Cluster 6: 20

Sugar Degradation 0.01654788849762285 2.0
Glycolysis 1.5272197664974587e-05 4.0

--------------------------------------------------------
Cluster 7: 14

Reactive Oxygen Specie Degradation 0.010602753145662433 2.0
Pyrimidine Nucleotide Salvage 0.0316311505098578 2.0

--------------------------------------------------------
Cluster 8: 4

Fermentation to Short-Chain Fatty Acids 0.0026914088224972953 2.0
Pyruvate Fermentation to Ethanol 0.0026914088224972953 2.0
TCA cycle 0.002809971767291062 2.0

--------------------------------------------------------
Cluster 9: 12

cob(II)yrinate <i>a,c</i>-diamide biosynthesis 0.01654788849762285 2.0
Assimilatory Sulfate Reduction 3.8368332270469076e-05 3.0

--------------------------------------------------------

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
  # Remove the CWD from sys.path while we load stuff.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.



       ␣
↪---------------------------------------------------------------------------


        IndexError                                Traceback (most recent call␣
↪last)


        <ipython-input-889-f6f70a2b4d9e> in <module>
         11     df=df.sort_values(by=['sort','pval'])
         12     plt.figure(figsize=(4,.5*len(df)))
    ---> 13     g = sns.
↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
         14     plt.ylabel('')
         15     plt.xlabel('Fraction of Annotated Genes')


        IndexError: list index out of range
```





```
<Figure size 288x108 with 0 Axes>
```

11 Enrichment using SEED terms

[633]:
```
print('SEED Pathway Terms')
terms=[]
for g in sets[('Regular', 'Cysteine-inhibition')]:
    vals=list(seed[seed.index==g]['pathway'].unique())
    terms+=vals
print('Cysteine: %s'%(len(pd.unique(terms))))

terms=[]
for g in sets[('Regular', 'Sulfide-inhibition')]:
    vals=list(seed[seed.index==g]['pathway'].unique())
    terms+=vals
print('Sulfide: %s'%(len(pd.unique(terms))))

print('Total unique terms: %s'%(len(seed['pathway'].unique())))
```

```
SEED Pathway Terms
Cysteine: 34
Sulfide: 31
Total unique terms: 108
```

[634]:
```
print('SEED Subpathway Terms')
terms=[]
for g in sets[('Regular', 'Cysteine-inhibition')]:
    vals=list(seed[seed.index==g]['subpathway'].unique())
    terms+=vals
print('Cysteine: %s'%(len(pd.unique(terms))))
```

```
terms=[]
for g in sets[('Regular', 'Sulfide-inhibition')]:
    vals=list(seed[seed.index==g]['subpathway'].unique())
    terms+=vals
print('Sulfide: %s'%(len(pd.unique(terms))))

print('Total unique terms: %s'%(len(seed['subpathway'].unique())))
```

```
SEED Subpathway Terms
Cysteine: 87
Sulfide: 65
Total unique terms: 545
```

[640]:

```
#SEED  pathway terms
pathways=seed['pathway'].reset_index()
pathways.columns=['ID','value']
clustdf=Enrichment(pathways,avgzvals,hits,y_pred,enrich='SEED Pathway')

#Add results to supplementary table
supp_table=pd.concat([supp_table,clustdf],ignore_index=True)

#Visualize significant hits using fraction of genes in each category found in␣
 ↪the cluster
for index, df in clustdf.groupby(by='clust'):
    df['sort']=[x.split()[-1] for x in df['name']]
    df=df.sort_values(by=['sort','pval'])
    plt.figure(figsize=(4,.5*len(df)))
    g = sns.
 ↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
    plt.ylabel('')
    plt.xlabel('Fraction of Annotated Genes')
    plt.xlim(0,1)
    #plt.savefig('Pathway-Clust%s-K-%s.pdf'%(index,k),bbox_inches='tight')
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  import sys
```

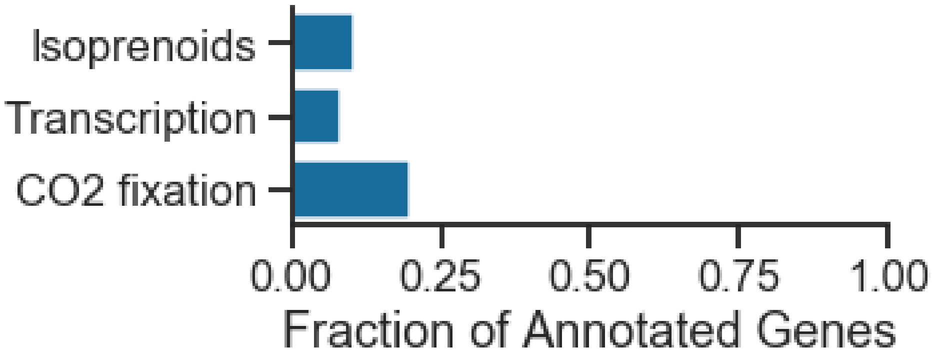
```
HBox(children=(FloatProgress(value=0.0, max=5.0), HTML(value='')))
```

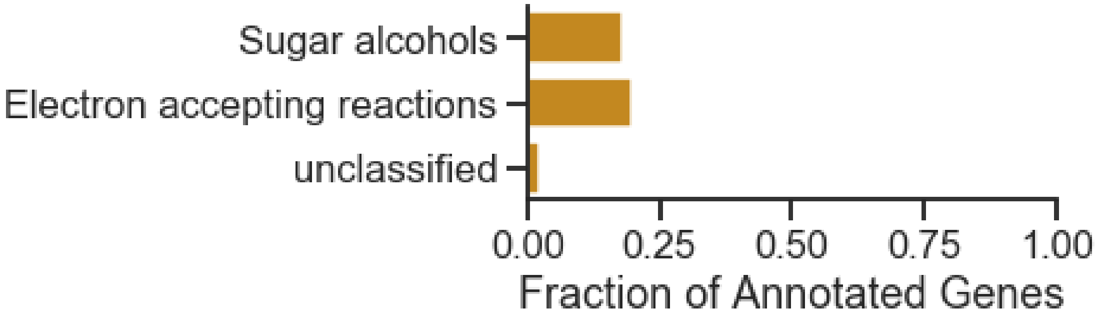
```
//anaconda3/lib/python3.7/site-packages/pandas/core/series.py:856:
RuntimeWarning: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

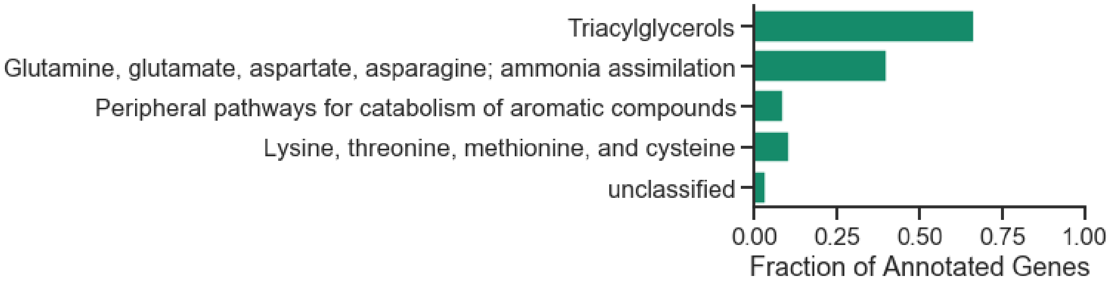
```
Cluster 1: 43
```

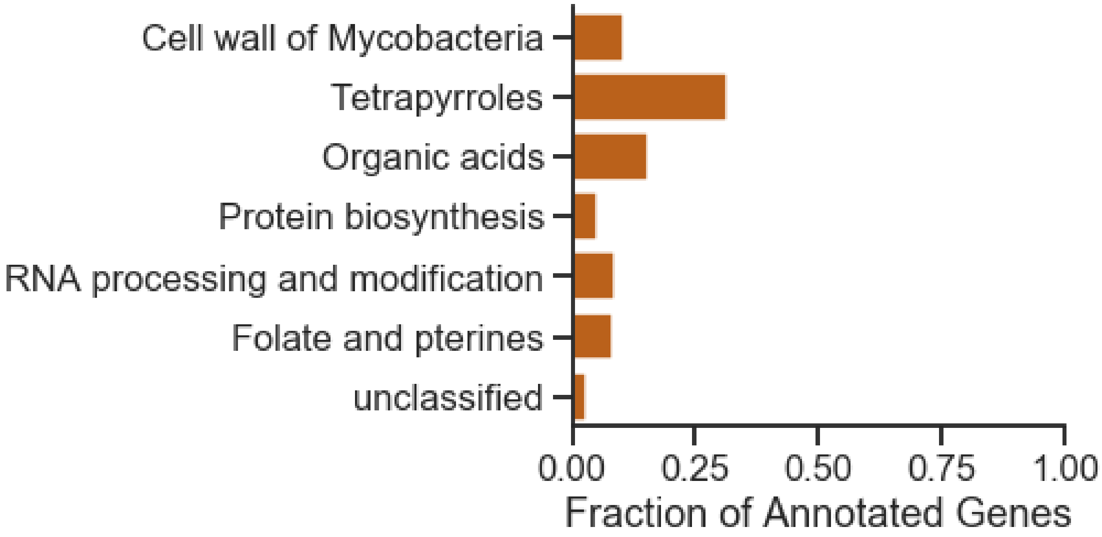
```
CO2 fixation 0.009436446601450581 2.0
Isoprenoids 0.023844537545838974 2.0
Transcription 0.030908593389290317 2.0

--------------------------------------------------------
Cluster 2: 64

unclassified 0.0 29.0
Electron accepting reactions 0.0035028666914867076 3.0
Sugar alcohols 0.023844537545838974 2.0

--------------------------------------------------------
Cluster 3: 124

unclassified 0.0 39.0
Triacylglycerols 0.0035028666914867076 2.0
Lysine, threonine, methionine, and cysteine 0.0045750459032774474 4.0
Glutamine, glutamate, aspartate, asparagine; ammonia assimilation
0.008974374307935847 2.0
Peripheral pathways for catabolism of aromatic compounds 0.024368582252021462
3.0

--------------------------------------------------------
Cluster 4: 90

unclassified 0.0 33.0
RNA processing and modification 0.009697894581988827 4.0
Cell wall of Mycobacteria 0.02006129940122972 3.0
Folate and pterines 0.030908593389290317 3.0
Organic acids 0.03532527388042957 2.0
Protein biosynthesis 0.03532527388042957 4.0
Tetrapyrroles 4.1157346788216925e-07 6.0

--------------------------------------------------------
Cluster 5: 41

unclassified 0.016362243407827708 12.0
One-carbon Metabolism 0.03532527388042957 2.0
Central carbohydrate metabolism 0.00011463005186538166 5.0

--------------------------------------------------------

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:65:
RuntimeWarning: divide by zero encountered in log
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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
```

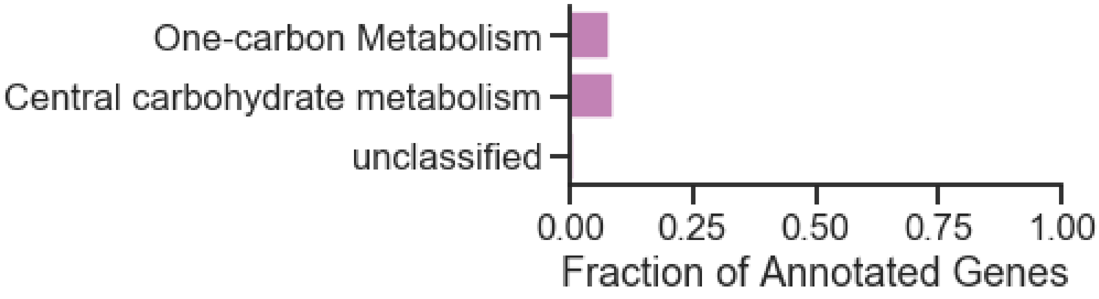
```
See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
SettingWithCopyWarning:
A value is 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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
```




Sugar alcohols
Electron accepting reactions
unclassified
0.00
0.25
0.50
0.75
1.00
Fraction of Annotated Genes


Triacylglycerols
Glutamine, glutamate, aspartate, asparagine; ammonia assimilation
Peripheral pathways for catabolism of aromatic compounds
Lysine, threonine, methionine, and cysteine
unclassified
0.00
0.25
0.50
0.75
1.00
Fraction of Annotated Genes


Cell wall of Mycobacteria
Tetrapyrroles
Organic acids
Protein biosynthesis
RNA processing and modification
Folate and pterines
unclassified
0.00
0.25
0.50
0.75
1.00
Fraction of Annotated Genes

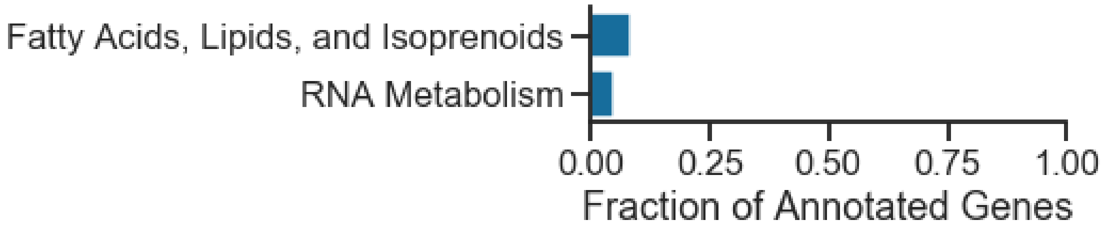


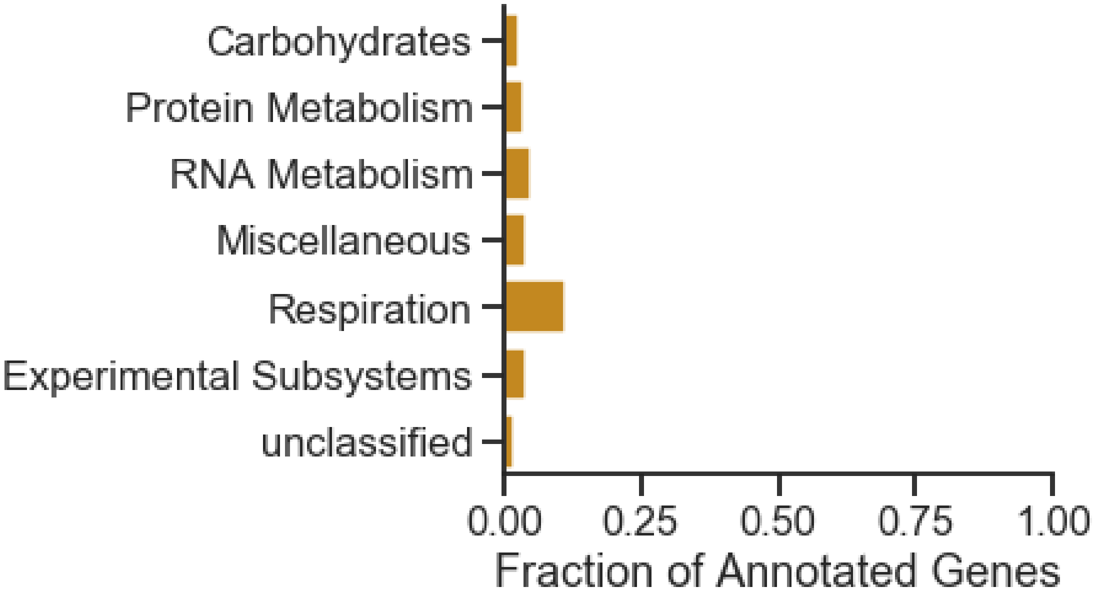
```
[641]: #SEED  superpathway terms
pathways=seed['superpathway'].reset_index()
pathways.columns=['ID','value']
clustdf=Enrichment(pathways,avgzvals,hits,y_pred,enrich='SEED Superpathway')

#Add results to supplementary table
supp_table=pd.concat([supp_table,clustdf],ignore_index=True)

#Visualize significant hits using fraction of genes in each category found in␣
 ↪the cluster
for index, df in clustdf.groupby(by='clust'):
    df['sort']=[x.split()[-1] for x in df['name']]
    df=df.sort_values(by=['sort','pval'])
    plt.figure(figsize=(4,.5*len(df)))
    g = sns.
 ↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
    plt.ylabel('')
    plt.xlabel('Fraction of Annotated Genes')
    plt.xlim(0,1)
    #plt.savefig('Pathway-Clust%s-K-%s.pdf'%(index,k),bbox_inches='tight')
```

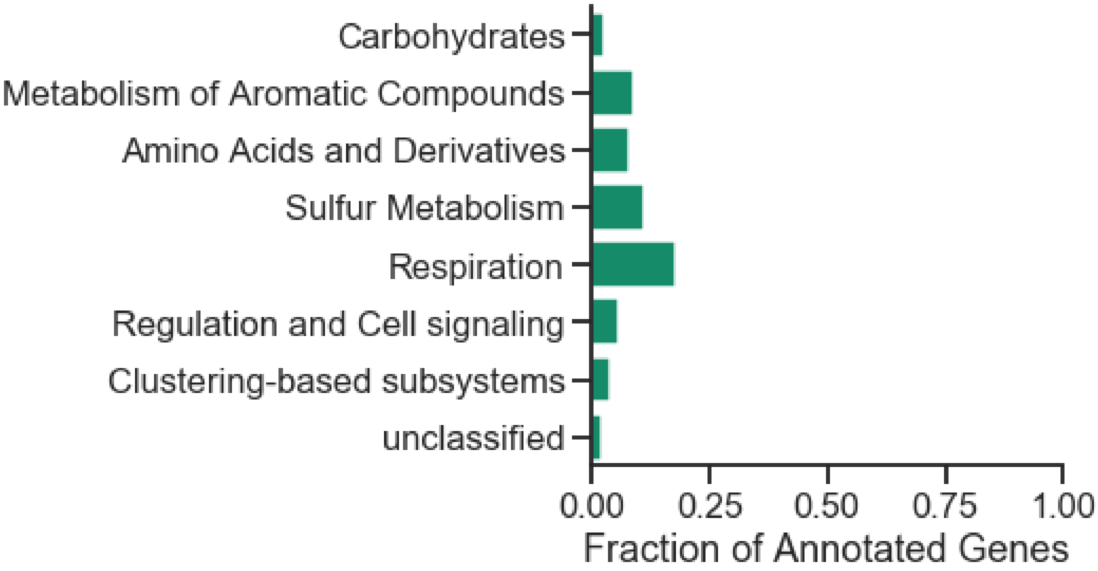
```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  import sys

HBox(children=(FloatProgress(value=0.0, max=5.0), HTML(value='')))



Cluster 1: 43

Fatty Acids, Lipids, and Isoprenoids 0.0037691526125920517 3.0
RNA Metabolism 0.014892942380968385 3.0

--------------------------------------------------------
```

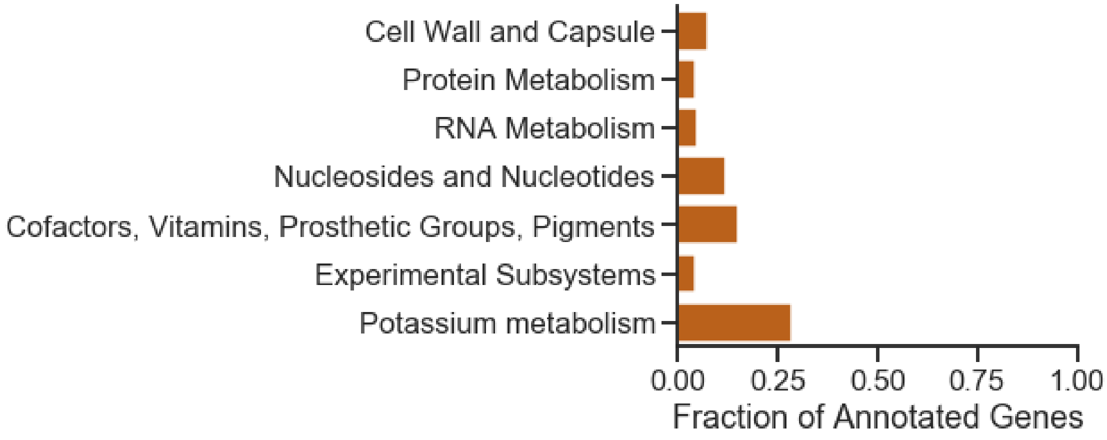
```
Cluster 2: 64

unclassified 0.0084425108586455 9.0
Protein Metabolism 0.019039329287081404 4.0
Carbohydrates 0.019846123150527587 5.0
RNA Metabolism 0.023550998604886037 3.0
Miscellaneous 0.03597678183045987 3.0
Experimental Subsystems 1.6383047410584841e-09 14.0
Respiration 4.904323797158241e-05 5.0

--------------------------------------------------------
Cluster 3: 124

unclassified 0.0027357536132126084 11.0
Regulation and Cell signaling 0.024833122922962067 3.0
Sulfur Metabolism 0.025700098133415002 2.0
Carbohydrates 0.03702916790564396 5.0
Amino Acids and Derivatives 4.4714092403878573e-07 9.0
Respiration 6.447804947064562e-09 8.0
Clustering-based subsystems 0.00028741868873017733 8.0
Metabolism of Aromatic Compounds 8.315426702869643e-06 7.0

--------------------------------------------------------
Cluster 4: 90

Nucleosides and Nucleotides 0.006165314413773915 3.0
Potassium metabolism 0.007541131288698269 2.0
Protein Metabolism 0.010160203893858783 5.0
RNA Metabolism 0.041746365323716075 3.0
Experimental Subsystems 4.962498211577822e-09 16.0
Cofactors, Vitamins, Prosthetic Groups, Pigments 1.6383047410584841e-09 10.0
Cell Wall and Capsule 0.00028741868873017733 6.0

--------------------------------------------------------
Cluster 5: 41

unclassified 0.015272013886471656 7.0
Amino Acids and Derivatives 0.037657683705838114 3.0
Carbohydrates 2.8723253902048134e-06 8.0

--------------------------------------------------------

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
```

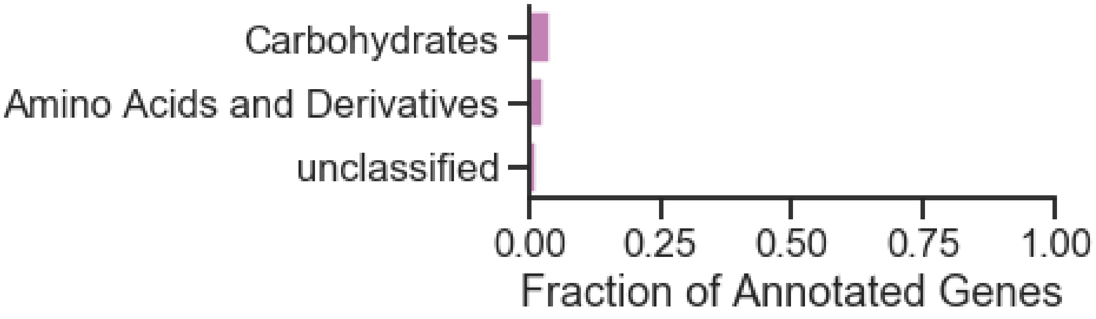
```
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
```




Carbohydrates
Protein Metabolism
RNA Metabolism
Miscellaneous
Respiration
Experimental Subsystems
unclassified
0.00
0.25
0.50
0.75
1.00
Fraction of Annotated Genes


Carbohydrates
Metabolism of Aromatic Compounds
Amino Acids and Derivatives
Sulfur Metabolism
Respiration
Regulation and Cell signaling
Clustering-based subsystems
unclassified
0.00
0.25
0.50
0.75
1.00
Fraction of Annotated Genes

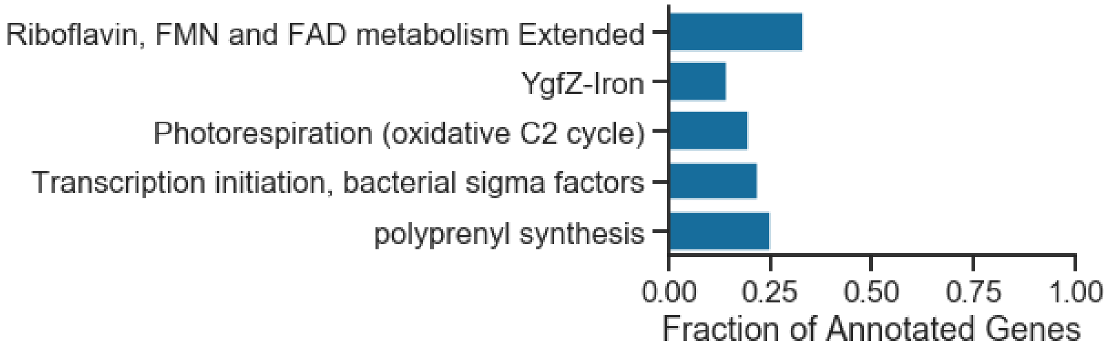


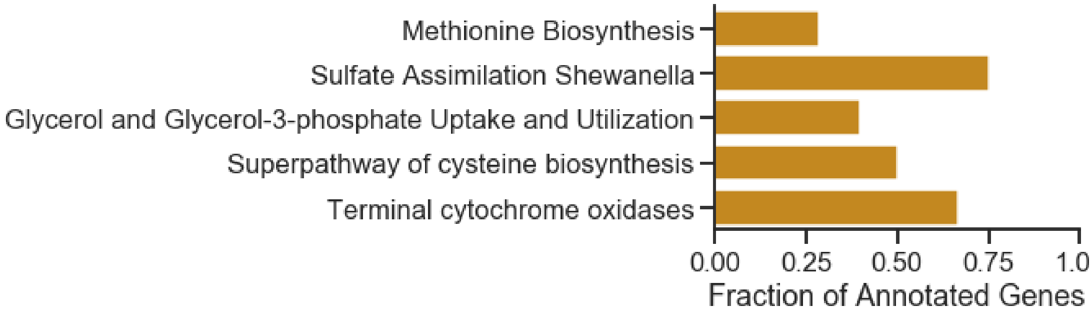


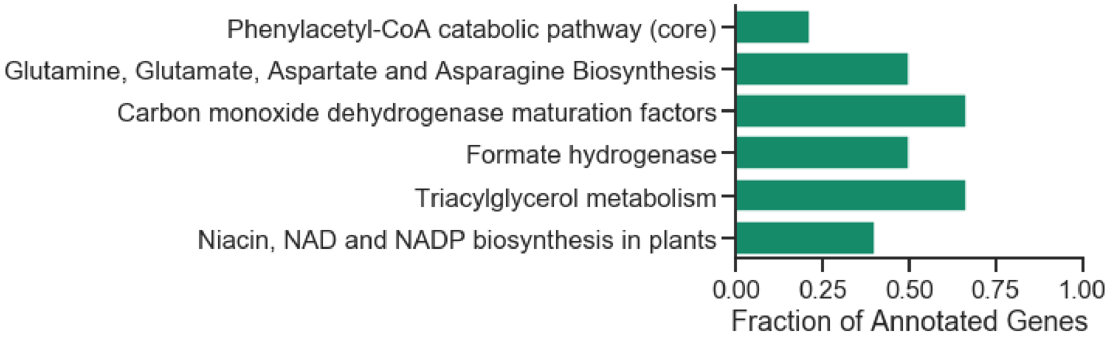
```
[642]: #SEED  subpathway terms
pathways=seed['subpathway'].reset_index()
pathways.columns=['ID','value']
clustdf=Enrichment(pathways,avgzvals,hits,y_pred,enrich='SEED subpathway')

#Add results to supplementary table
supp_table=pd.concat([supp_table,clustdf],ignore_index=True)

#Visualize significant hits using fraction of genes in each category found in␣
 ↪the cluster
for index, df in clustdf.groupby(by='clust'):
    df['sort']=[x.split()[-1] for x in df['name']]
    df=df.sort_values(by=['sort','pval'])
    plt.figure(figsize=(4,.5*len(df)))
    g = sns.
 ↪barplot(y='name',x='percent',data=df,color=colors[int(index)-1],dodge=False)
    plt.ylabel('')
    plt.xlabel('Fraction of Annotated Genes')
    plt.xlim(0,1)
    #plt.savefig('Pathway-Clust%s-K-%s.pdf'%(index,k),bbox_inches='tight')
```

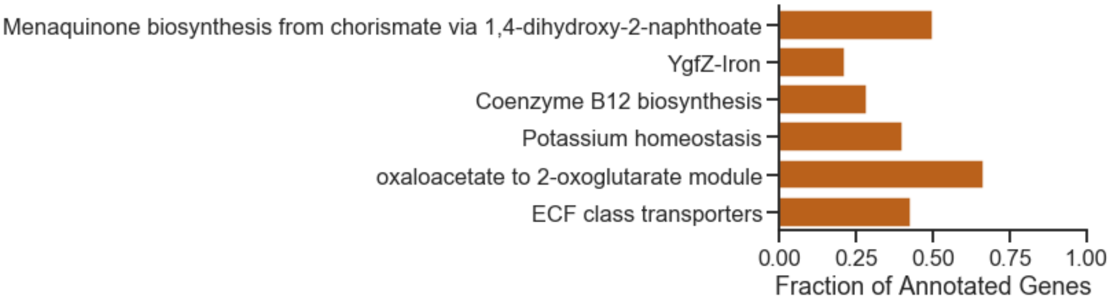
```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7:
TqdmDeprecationWarning: This function will be removed in tqdm==5.0.0
Please use `tqdm.notebook.tqdm` instead of `tqdm.tqdm_notebook`
  import sys

HBox(children=(FloatProgress(value=0.0, max=5.0), HTML(value='')))



Cluster 1: 43

Riboflavin, FMN and FAD metabolism Extended 0.024101607878215275 2.0
polyprenyl synthesis 0.02656413226062183 2.0
Transcription initiation, bacterial sigma factors 0.02656413226062183 2.0
Photorespiration (oxidative C2 cycle) 0.02656413226062183 2.0
YgfZ-Iron 0.04011415960418026 2.0

--------------------------------------------------------
Cluster 2: 64

Terminal cytochrome oxidases 0.024101607878215275 2.0
Superpathway of cysteine biosynthesis 0.02656413226062183 2.0
Glycerol and Glycerol-3-phosphate Uptake and Utilization 0.02656413226062183 2.0
Methionine Biosynthesis 0.042119677230337445 2.0
Sulfate Assimilation Shewanella 0.001600695915684269 3.0

--------------------------------------------------------
Cluster 3: 124

Triacylglycerol metabolism 0.024416641478906747 2.0
Carbon monoxide dehydrogenase maturation factors 0.024416641478906747 2.0
Formate hydrogenase 0.02656413226062183 2.0
Glutamine, Glutamate, Aspartate and Asparagine Biosynthesis 0.02656413226062183
2.0
Phenylacetyl-CoA catabolic pathway (core) 0.02656413226062183 3.0
Niacin, NAD and NADP biosynthesis in plants 0.03333083923704433 2.0

--------------------------------------------------------
Cluster 4: 90

Coenzyme B12 biosynthesis 0.009890398679201351 4.0
ECF class transporters 0.013356720328933442 3.0
Menaquinone biosynthesis from chorismate via 1,4-dihydroxy-2-naphthoate
0.02656413226062183 2.0
YgfZ-Iron 0.027555095146706843 3.0
Potassium homeostasis 0.035447477416043434 2.0
```

```
oxaloacetate to 2-oxoglutarate module 0.0005095473057407028 4.0

--------------------------------------------------------

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:11:
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: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
```









12 Gene level comparisons for select pathways

- Genes from pathways selected from MetaCyc, GO, and SEED are compared for differential expression
- Welch's t-test is used, genes are not assumed to have equal variance
- p-values are corrected using FDR method
- Results are compiled into Supplementary Table 2

```
[539]: def get_gene_names(genes,seed,blast_annot):
           #Function which extracts gene annotation from KIP7 (BLAST annotations) and␣
        ↪SEED (DIAMOND annotations)
           names=[]
           for g in genes:
               if g in blast_annot.index:
                   name=blast_annot.loc[g,'C5.transcript.description']
                   if type(name) != type(''):
                       name=name.iloc[0]
                   names.append(name)

               elif g in seed.index:
                   name=seed.loc[g,'function']
                   if type(name) != type(''):
                       name=name.iloc[0]
                   names.append(name)

               else:
                   names.append('Unknown Function')
           return names

       def pathway_degs(genes,zvals,pathway,gene_map,seed,blast_annot,p=0.
        ↪05,correction=True,cond=['Regular','Cysteine-inhibition','Sulfide-inhibition']):
        ↪
           #Gene level expression comparisons for specific sets of genes
           df=zvals.loc[genes].groupby(level=0,axis=1).mean()[cond]
           df.columns=['Avg Z-score Nt','Avg Z-score Cys','Avg Z-score Sulf']
           df['description']=get_gene_names(df.index,seed,blast_annot)
           df['Gene ID']=df.index.map(gene_map)
           df['pathway']=pathway
           svals=zvals.loc[genes]['Sulfide-inhibition']
           rvals=zvals.loc[genes]['Regular']
           cvals=zvals.loc[genes]['Cysteine-inhibition']

           #Test for differential expression
           cys_pvals=ttest_ind(cvals,rvals,equal_var=False,axis=1).pvalue
           sulf_pvals=ttest_ind(svals,rvals,equal_var=False,axis=1).pvalue
           cys_sulf_pvals=ttest_ind(svals,cvals,equal_var=False,axis=1).pvalue

           #Correct p-values using FDR
           if correction:
               cys_pvals=multipletests(cys_pvals, method='fdr_bh')[:2][1]
               sulf_pvals=multipletests(sulf_pvals, method='fdr_bh')[:2][1]
               cys_sulf_pvals=multipletests(cys_sulf_pvals, method='fdr_bh')[:2][1]

           #Save results
           df['Cys-vs-Nt p-value']=cys_pvals
```

```
    df['Sulf-vs-Nt p-value']=sulf_pvals
    df['Cys-vs-Sulf p-value']=cys_sulf_pvals

    #Report
    print(pathway)
    print('DEGs Down Cys vs Nt: %s'%(df[(df['Cys-vs-Nt p-value']<p)&(df['Avg␣
↪Z-score Cys']<df['Avg Z-score Nt'])].shape[0]))
    print('DEGs Up Cys vs Nt: %s'%(df[(df['Cys-vs-Nt p-value']<p)&(df['Avg␣
↪Z-score Cys']>df['Avg Z-score Nt'])].shape[0]))
    print('')
    print('DEGs Down Sulf vs Nt: %s'%(df[(df['Sulf-vs-Nt p-value']<p)&(df['Avg␣
↪Z-score Sulf']<df['Avg Z-score Nt'])].shape[0]))
    print('DEGs Up Sulf vs Nt: %s'%(df[(df['Sulf-vs-Nt p-value']<p)&(df['Avg␣
↪Z-score Sulf']>df['Avg Z-score Nt'])].shape[0]))
    print('')
    print('DEGs Down Cys vs Sulf: %s'%(df[(df['Sulf-vs-Nt p-value']<p)&(df['Avg␣
↪Z-score Cys']<df['Avg Z-score Sulf'])].shape[0]))
    print('DEGs Up Cys vs Sulf: %s'%(df[(df['Cys-vs-Sulf p-value']<p)&(df['Avg␣
↪Z-score Cys']>df['Avg Z-score Sulf'])].shape[0]))
    print('')
    print('Total Genes: %s'%(len(genes)))
    return df
```

[704]:
```
def pathway_boxplots(genes,title,zvals,save=False,filename=None,p=0.
↪05,cond=['Regular','Cysteine-inhibition','Sulfide-inhibition'],alpha=0.5):
    c=sns.color_palette('colorblind')
    #Get data and format for plotting
    df=zvals.reindex(genes).dropna().loc[:,cond]
    df=df.reset_index().melt(id_vars=['transcrip_name'])
    df.columns=['gene_id','condition','rep','time_point','zscore']

    #Create Figure
    plt.figure(figsize=(4,2.5))
    #Make boxplot and overlay with scatter points
    g=sns.
↪boxplot(y='condition',x='zscore',data=df,palette=[c[0],c[2],c[4]],order=['Regular','Cystein
    sns.
↪stripplot(y='condition',x='zscore',data=df,color='black',alpha=alpha,order=['Regular','Cyst
    g.set_title(title+' ($n=$%s)'%(int(len(genes))))

    #Differential expression test (pathway level), Welch's t-test
    cys=df[df.condition=='Cysteine-inhibition'].zscore.values
    reg=df[df.condition=='Regular'].zscore.values
    sulf=df[df.condition=='Sulfide-inhibition'].zscore.values
    p_sulf=ttest_ind(reg,sulf,equal_var=False).pvalue
    p_cys=ttest_ind(reg,cys,equal_var=False).pvalue
    p_cys_sulf=ttest_ind(cys,sulf,equal_var=False).pvalue

    #Print Results
    print('Cys vs Nt: %s'%(p_cys))
    print('Sulf vs Nt: %s'%(p_sulf))
    print('Cys vs Sulf: %s'%(p_cys_sulf))

    #Add bars if comparisons are significant
    if p_cys < p:
        g.plot((2.5,2.5),(0,1),'k',linewidth=2,linestyle='-')
    if p_sulf < p:
        g.plot((2.7,2.7),(0,2),'k',linewidth=2,linestyle='-')
    if p_cys_sulf < p:
        g.plot((2.9,2.9),(1,2),'k',linewidth=2,linestyle='-')

    #Add labels
    g.set_ylabel('')
    g.set_xlabel('Z-Score')
    g.set_xlim(-4.5,4.5)
    g.set_xticks([-4,-2,0,2,4])
    g.set_yticklabels(['No treatment','Cysteine treatment', 'Sulfide␣
 ↪treatment'])
    if save:
        if filename != None:
            plt.savefig('%s.pdf'%(filename),bbox_inches='tight')
            plt.savefig('%s.png'%(filename),bbox_inches='tight',dpi=500)
        else:
            plt.savefig('%s.pdf'%(title),bbox_inches='tight')
            plt.savefig('%s.png'%(title),bbox_inches='tight',dpi=500)
    plt.show()
    plt.clf()

def␣
 ↪clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed,names=False,save=False,filename='hea
 ↪5,cond=['Regular','Cysteine-inhibition','Sulfide-inhibition']):
    #Get data
    df=avgzvals.reindex(genes).dropna()[cond]
    #Scale figure size to dimensions of data table
    rows=df.shape[0]*scale
    cols=df.shape[1]*scale
    #If no genes names are provided get them
    if type(names)!=type([]):
        names = get_gene_names(genes,seed,blast_annot)
    #Add gene ID to gene name
    df.index=[gene_map[df.index[x]]+','+names[x] for x in range(len(df.index))]
    #Create heatmap
    clust = sns.clustermap(df,cmap='RdYlBu',col_cluster=False,vmax=-2.5,vmin=2.
↪5,figsize=(cols,rows),yticklabels=1)
    clust.ax_heatmap.set(ylabel='',xlabel='',xticks=[4*x+2 for x in␣
↪range(len(cond))])
    clust.ax_heatmap.set_xticklabels([x.split('-')[0] for x in cond],rotation=0)
    clust.cax.set_visible(False)
    #Save if desired
    if save:
        plt.savefig('%s.pdf'%(filename),bbox_inches='tight')
        plt.savefig('%s.png'%(filename),bbox_inches='tight',dpi=500)
```

[1009]:

```
supp_table2=pd.DataFrame() #Initialized supplementary table 2
```

[1010]:

```
#Sulfur, cysteine, and methione pathways
S_paths=pd.DataFrame()
names=sorted(path_annot.value.dropna().unique())
for name in names:
    if ('cysteine' in name.lower() or 'sulfur' in name.lower() or 'sulfide' in␣
↪name.lower() or 'sulfate' in name.lower() or 'methionine' in name.lower())␣
↪and 'adenosyl' not in name.lower() and 'iron-sulfur' not in name.lower():
        df=path_annot[path_annot['value']==name]
        S_paths=pd.concat([S_paths,df])
res=pd.DataFrame()
for path,s_df in S_paths.groupby(by='value'):
    genes=s_df.ID.unique()
    df=pathway_degs(genes,zvals,'MetaCyc '+path,gene_map,seed,blast_annot)
    res=pd.concat([res,df])
    print('---------------')

temp=res.drop_duplicates('Gene ID').copy()
supp_table2=pd.concat([supp_table2,temp])
temp
```

```
MetaCyc Assimilatory Sulfate Reduction
DEGs Down Cys vs Nt: 4
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 5
---------------
MetaCyc L-cysteine Biosynthesis
```

```
DEGs Down Cys vs Nt: 4
DEGs Up Cys vs Nt: 2

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 9
--------------
MetaCyc L-cysteine Degradation
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 1

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 2

Total Genes: 2
--------------
MetaCyc L-cysteine biosynthesis I
DEGs Down Cys vs Nt: 2
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 2
--------------
MetaCyc L-cysteine biosynthesis III (from L-homocysteine)
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 2

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 3
--------------
MetaCyc L-cysteine biosynthesis VI (from L-methionine)
```

```
DEGs Down Cys vs Nt: 8
DEGs Up Cys vs Nt: 3

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 4

Total Genes: 12
--------------
MetaCyc L-cysteine degradation II
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 2

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 3
--------------
MetaCyc L-homocysteine Degradation
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 1

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 2

Total Genes: 2
--------------
MetaCyc L-homocysteine biosynthesis
DEGs Down Cys vs Nt: 2
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 2
--------------
MetaCyc L-methionine <i>De Novo</i> Biosynthesis
```

```
DEGs Down Cys vs Nt: 9
DEGs Up Cys vs Nt: 2

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 2

Total Genes: 11
--------------
MetaCyc L-methionine Degradation
DEGs Down Cys vs Nt: 1
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 2
--------------
MetaCyc L-methionine biosynthesis II (plants)
DEGs Down Cys vs Nt: 15
DEGs Up Cys vs Nt: 3

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 18
--------------
MetaCyc L-methionine biosynthesis III
DEGs Down Cys vs Nt: 5
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 5
--------------
MetaCyc L-methionine degradation I (to L-homocysteine)
```

```
DEGs Down Cys vs Nt: 2
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 3
--------------
MetaCyc Thiosulfate Disproportionation
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 1
--------------
MetaCyc assimilatory sulfate reduction III
DEGs Down Cys vs Nt: 4
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 5
--------------
MetaCyc homocysteine and cysteine interconversion
DEGs Down Cys vs Nt: 9
DEGs Up Cys vs Nt: 3

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 4

Total Genes: 13
--------------
MetaCyc superpathway of L-cysteine biosynthesis (mammalian)
```

```
DEGs Down Cys vs Nt: 2
DEGs Up Cys vs Nt: 2

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 6
--------------
MetaCyc thiosulfate disproportionation IV (rhodanese)
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 1

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 1

Total Genes: 3
--------------
```

```
[1010]:                   Avg Z-score Nt  Avg Z-score Cys  Avg Z-score Sulf  \
transcrip_name
Ga0242637_112608        0.271666         0.052816          0.364231
Ga0242637_11954         1.059089        -0.819600          0.703433
Ga0242637_11953         0.970884        -0.644325          0.759231
Ga0242637_113589        1.109759        -0.952780          0.768441
Ga0242637_113588        0.999255        -0.700830          0.848884
Ga0242637_111432        0.755629        -0.703066          0.630829
Ga0242637_112372        0.226274         0.248907          0.105027
Ga0242637_11791         0.415937         0.885212         -0.270305
Ga0242637_11841        -0.658125         0.937829         -0.659650
Ga0242637_111070        1.038068        -0.547347          0.453216
Ga0242637_11110         0.879477        -0.224329          0.706209
Ga0242637_113154       -0.649825         0.073243         -0.568977
Ga0242637_111881        0.555394        -0.491054          0.606249
Ga0242637_112177        0.423704        -0.107622          0.672289
Ga0242637_113647       -0.465028         0.120405         -0.681814
Ga0242637_11842        -0.302314         0.951142         -0.386102
Ga0242637_111885        0.606530        -0.882611          0.342718
Ga0242637_111884        0.577671        -0.579410          0.318021
Ga0242637_111883        0.607412        -0.562052          0.572518
Ga0242637_111882        0.456469        -0.417048          0.599365
Ga0242637_111433        0.208098        -0.864156          0.479420
```

```
Ga0242637_111318        0.856807        -0.314195          0.332153
Ga0242637_111579        0.592622        -0.772198          0.778421
Ga0242637_112246        0.696901        -0.622394          0.255846
Ga0242637_112310        0.647295        -0.946562          0.614213
Ga0242637_111313        0.269856        -0.804675          0.131942
Ga0242637_111816        0.562884        -0.381821          0.090555
Ga0242637_112312        0.682140        -0.627252          0.687592
Ga0242637_112311        0.693493        -0.872903          0.617582
Ga0242637_111815        0.519122        -0.295656          0.019220
Ga0242637_111580        0.596825        -1.125751          0.652221
Ga0242637_111032       -0.126455        -0.273852         -0.394681
Ga0242637_111033       -0.494189         0.436511         -0.685269
Ga0242637_111031       -0.149764        -0.401802         -0.272072

                                                        description  \
transcrip_name
Ga0242637_112608    translation elongation factor 1A (EF-1A/EF-Tu)
Ga0242637_11954              sulfate adenylyltransferase subunit 1
Ga0242637_11953              sulfate adenylyltransferase subunit 2
Ga0242637_113589    phosphoadenylylsulfate reductase (thioredoxin)
Ga0242637_113588                     sulfite reductase (ferredoxin)
Ga0242637_111432                    methionine adenosyltransferase
Ga0242637_112372                            adenosylhomocysteinase
Ga0242637_11791                         cystathionine beta-synthase
Ga0242637_11841                         cystathionine gamma-synthase
Ga0242637_111070                  O-acetylhomoserine sulfhydrylase
Ga0242637_11110                                   cysteine synthase
Ga0242637_113154  cystathionine gamma-synthase/methionine-gamma-…
Ga0242637_111881                 O-succinylhomoserine sulfhydrylase
Ga0242637_112177                       cystathionine gamma-synthase
Ga0242637_113647                                      tryptophanase
Ga0242637_11842           uncharacterized protein (TIGR01319 family)
Ga0242637_111885  dinuclear metal center YbgI/SA1388 family protein
Ga0242637_111884                               hypothetical protein
Ga0242637_111883                   probable phosphoglycerate mutase
Ga0242637_111882                rhodanese-related sulfurtransferase
Ga0242637_111433             replication restart DNA helicase PriA
Ga0242637_111318                    homoserine O-acetyltransferase
Ga0242637_111579               methionine synthase (B12-dependent)
Ga0242637_112246  cobalamin-independent methionine synthase cata…
Ga0242637_112310                                 homoserine kinase
Ga0242637_111313                           cystathionine beta-lyase
Ga0242637_111816  menaquinol-cytochrome c reductase cytochrome c…
Ga0242637_112312                          homoserine dehydrogenase
Ga0242637_112311                              L-threonine synthase
Ga0242637_111815                     cytochrome c oxidase subunit 3
Ga0242637_111580       HAD superfamily hydrolase (TIGR01509 family)
```

```
Ga0242637_111032   thiosulfate/3-mercaptopyruvate sulfurtransferase
Ga0242637_111033         DNA-binding transcriptional MerR regulator
Ga0242637_111031                cysteine desulfuration protein SufE

                    Gene ID  \
transcrip_name
Ga0242637_112608  Ical_1745
Ga0242637_11954   Ical_3638
Ga0242637_11953   Ical_3637
Ga0242637_113589  Ical_2819
Ga0242637_113588  Ical_2818
Ga0242637_111432  Ical_0461
Ga0242637_112372  Ical_1486
Ga0242637_11791   Ical_3459
Ga0242637_11841   Ical_3515
Ga0242637_111070  Ical_0077
Ga0242637_11110   Ical_0109
Ga0242637_113154  Ical_2341
Ga0242637_111881  Ical_0948
Ga0242637_112177  Ical_1273
Ga0242637_113647  Ical_2884
Ga0242637_11842   Ical_3516
Ga0242637_111885  Ical_0952
Ga0242637_111884  Ical_0951
Ga0242637_111883  Ical_0950
Ga0242637_111882  Ical_0949
Ga0242637_111433  Ical_0462
Ga0242637_111318  Ical_0338
Ga0242637_111579  Ical_0620
Ga0242637_112246  Ical_1348
Ga0242637_112310  Ical_1419
Ga0242637_111313  Ical_0334
Ga0242637_111816  Ical_0877
Ga0242637_112312  Ical_1421
Ga0242637_112311  Ical_1420
Ga0242637_111815  Ical_0876
Ga0242637_111580  Ical_0622
Ga0242637_111032  Ical_0035
Ga0242637_111033  Ical_0036
Ga0242637_111031  Ical_0034

                                                            pathway  \
transcrip_name
Ga0242637_112608             MetaCyc Assimilatory Sulfate Reduction
Ga0242637_11954              MetaCyc Assimilatory Sulfate Reduction
Ga0242637_11953              MetaCyc Assimilatory Sulfate Reduction
Ga0242637_113589             MetaCyc Assimilatory Sulfate Reduction
```

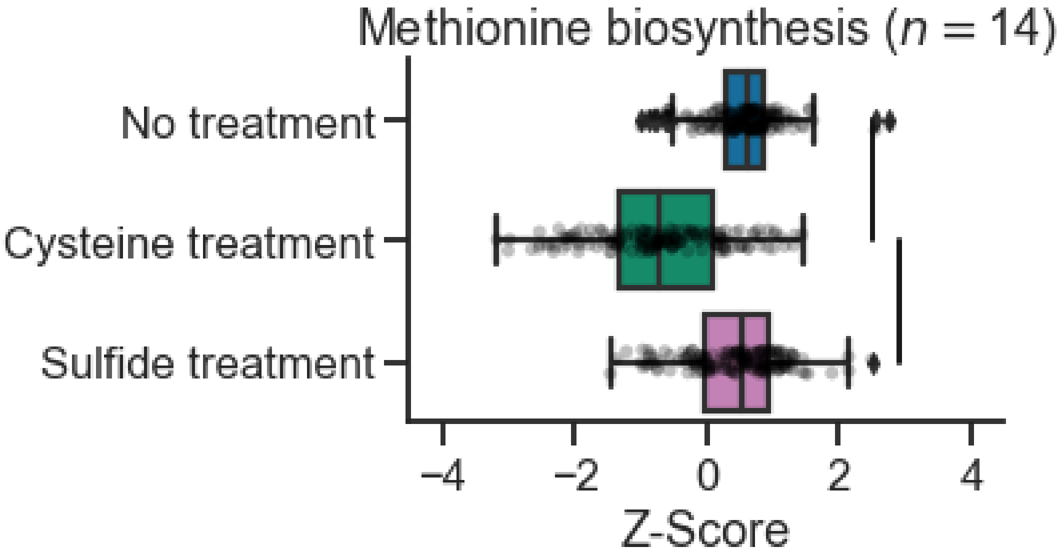
```
Ga0242637_113588             MetaCyc Assimilatory Sulfate Reduction
Ga0242637_111432                    MetaCyc L-cysteine Biosynthesis
Ga0242637_112372                    MetaCyc L-cysteine Biosynthesis
Ga0242637_11791                     MetaCyc L-cysteine Biosynthesis
Ga0242637_11841                     MetaCyc L-cysteine Biosynthesis
Ga0242637_111070                    MetaCyc L-cysteine Biosynthesis
Ga0242637_11110                     MetaCyc L-cysteine Biosynthesis
Ga0242637_113154                    MetaCyc L-cysteine Biosynthesis
Ga0242637_111881                    MetaCyc L-cysteine Biosynthesis
Ga0242637_112177                    MetaCyc L-cysteine Biosynthesis
Ga0242637_113647                     MetaCyc L-cysteine Degradation
Ga0242637_11842   MetaCyc L-cysteine biosynthesis III (from L-ho…
Ga0242637_111885  MetaCyc L-cysteine biosynthesis VI (from L-met…
Ga0242637_111884  MetaCyc L-cysteine biosynthesis VI (from L-met…
Ga0242637_111883  MetaCyc L-cysteine biosynthesis VI (from L-met…
Ga0242637_111882  MetaCyc L-cysteine biosynthesis VI (from L-met…
Ga0242637_111433  MetaCyc L-cysteine biosynthesis VI (from L-met…
Ga0242637_111318                MetaCyc L-homocysteine biosynthesis
Ga0242637_111579   MetaCyc L-methionine <i>De Novo</i> Biosynthesis
Ga0242637_112246   MetaCyc L-methionine <i>De Novo</i> Biosynthesis
Ga0242637_112310   MetaCyc L-methionine <i>De Novo</i> Biosynthesis
Ga0242637_111313   MetaCyc L-methionine <i>De Novo</i> Biosynthesis
Ga0242637_111816   MetaCyc L-methionine <i>De Novo</i> Biosynthesis
Ga0242637_112312      MetaCyc L-methionine biosynthesis II (plants)
Ga0242637_112311      MetaCyc L-methionine biosynthesis II (plants)
Ga0242637_111815      MetaCyc L-methionine biosynthesis II (plants)
Ga0242637_111580      MetaCyc L-methionine biosynthesis II (plants)
Ga0242637_111032             MetaCyc Thiosulfate Disproportionation
Ga0242637_111033  MetaCyc thiosulfate disproportionation IV (rho…
Ga0242637_111031  MetaCyc thiosulfate disproportionation IV (rho…

                  Cys-vs-Nt p-value  Sulf-vs-Nt p-value  Cys-vs-Sulf p-value
transcrip_name
Ga0242637_112608       3.317187e-01            0.799897         3.938958e-01
Ga0242637_11954        3.279869e-09            0.340392         9.618885e-06
Ga0242637_11953        1.857204e-08            0.536017         3.934473e-05
Ga0242637_113589       3.214454e-11            0.340392         7.175715e-07
Ga0242637_113588       7.076690e-12            0.536017         7.175715e-07
Ga0242637_111432       1.532323e-04            0.550646         3.226902e-04
Ga0242637_112372       9.652999e-01            0.881606         7.485490e-01
Ga0242637_11791        1.466911e-01            0.201220         2.767005e-04
Ga0242637_11841        6.195838e-04            0.984983         4.904501e-04
Ga0242637_111070       5.703322e-06            0.201220         2.767005e-04
Ga0242637_11110        7.604432e-05            0.550646         4.689949e-04
Ga0242637_113154       5.677283e-06            0.550646         7.378830e-05
Ga0242637_111881       3.146723e-06            0.881606         7.378830e-05
Ga0242637_112177       5.065243e-02            0.550646         1.645893e-02
```

```
Ga0242637_113647        7.913917e-02          0.984983          3.513160e-02
Ga0242637_11842         2.854598e-02          0.366974          1.359835e-02
Ga0242637_111885        5.732314e-03          0.562074          1.596018e-02
Ga0242637_111884        2.347856e-02          0.562074          6.086063e-02
Ga0242637_111883        1.471586e-02          0.973393          1.581440e-02
Ga0242637_111882        1.858662e-04          0.562074          6.558960e-05
Ga0242637_111433        5.732314e-03          0.705372          2.266585e-03
Ga0242637_111318        4.540535e-04          0.074885          4.226878e-02
Ga0242637_111579        8.485023e-05          0.576868          6.012380e-05
Ga0242637_112246        4.027364e-03          0.241647          4.839027e-02
Ga0242637_112310        1.521570e-03          0.888190          2.065073e-03
Ga0242637_111313        4.117332e-03          0.888190          4.839027e-02
Ga0242637_111816        4.785920e-03          0.206000          1.490355e-01
Ga0242637_112312        5.475576e-04          0.984983          5.254436e-04
Ga0242637_112311        1.100440e-03          0.854469          1.577865e-03
Ga0242637_111815        4.664390e-03          0.482641          2.480843e-01
Ga0242637_111580        1.100440e-03          0.968934          9.809003e-04
Ga0242637_111032        6.864410e-01          0.363447          7.280936e-01
Ga0242637_111033        1.024785e-02          0.559958          1.747628e-03
Ga0242637_111031        6.864410e-01          0.641223          7.760039e-01
```

[1011]:

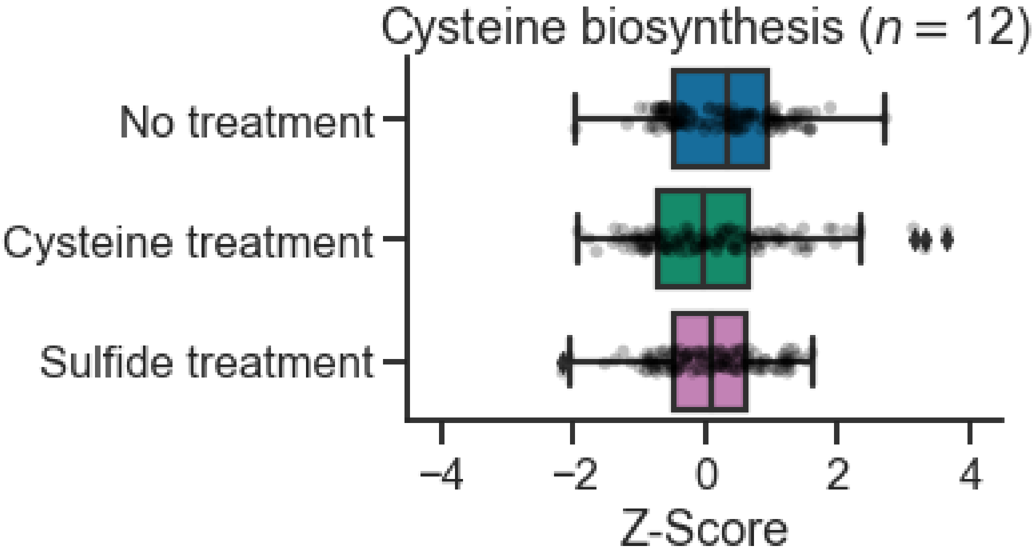
```
meth = []
cys =[]
for i in temp.index.values:
    if 'methionine' in temp.loc[i,'pathway'].lower():
        meth.append(i)
    elif 'cysteine' in temp.loc[i,'pathway'].lower():
        cys.append(i)
pathway_boxplots(meth,'Methionine biosynthesis',zvals,alpha=0.25,save=True)
pathway_boxplots(cys,'Cysteine biosynthesis',zvals,alpha=0.25)
```

```
Cys vs Nt: 6.159240372543156e-35
Sulf vs Nt: 0.10741412544489216
Cys vs Sulf: 5.534299608336247e-28
```



```
Cys vs Nt: 0.08386505326991607
Sulf vs Nt: 0.05295855077713129
Cys vs Sulf: 0.9309137066121236
```

```
<Figure size 432x288 with 0 Axes>
```

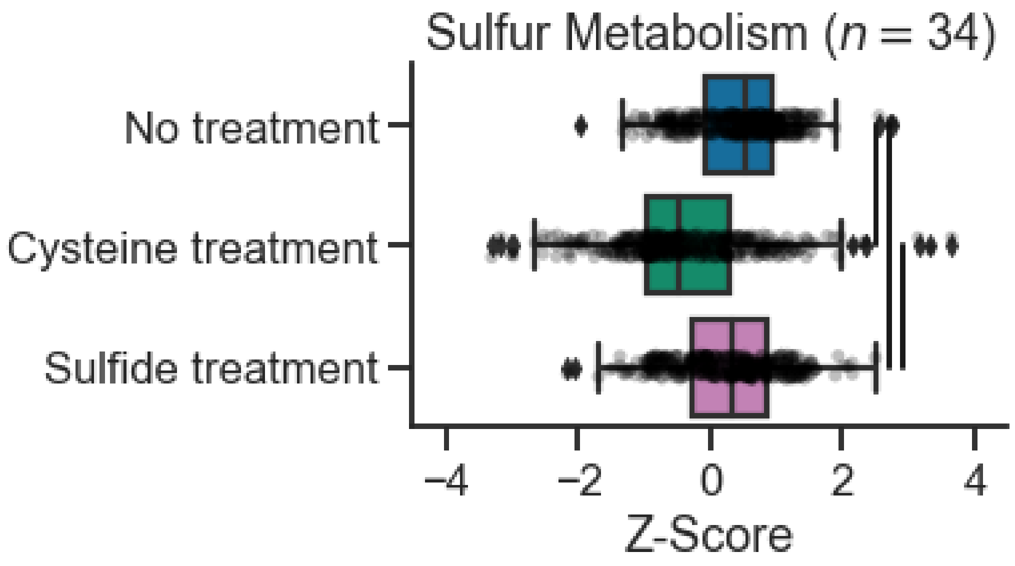


```
<Figure size 432x288 with 0 Axes>
```

[1012]:

```
#Closer looks at Cysteine and methione vs other sulfur metabolism pathways
temp=res.drop_duplicates('Gene ID').copy()
print('DEGs Cys all pathways: %s'%(temp[(temp['Cys-vs-Nt p-value']<0.
 ↪05)&(temp['Avg Z-score Cys']<temp['Avg Z-score Nt'])].shape[0]))
print('DEGs Sulf all pathways: %s'%(temp[temp['Sulf-vs-Nt p-value']<0.05].
 ↪shape[0]))
print('Pathway Genes: %s'%(temp.shape[0]))
print('------------')
idx=[]
for i in res.index:
    if 'l-cysteine biosynthesis' in str(res.loc[i,'pathway']).lower() or␣
 ↪'l-methionine' in str(res.loc[i,'pathway']).lower():
        idx.append(i)
temp=res.loc[idx].drop_duplicates('Gene ID').copy()
print('DEGs Cys cys and met pathways: %s'%(temp[(temp['Cys-vs-Nt p-value']<0.
 ↪05)&(temp['Avg Z-score Cys']<temp['Avg Z-score Nt'])].shape[0]))
print('DEGs Sulf cys and met pathways: %s'%(temp[temp['Sulf-vs-Nt p-value']<0.
 ↪05].shape[0]))
print('Cys and met pathway genes: %s'%(temp.shape[0]))
print('------------')
print('Pathway level comparison')
pathway_boxplots(S_paths.ID.unique(),'Sulfur Metabolism',zvals,alpha=0.
 ↪25,save=True)
```

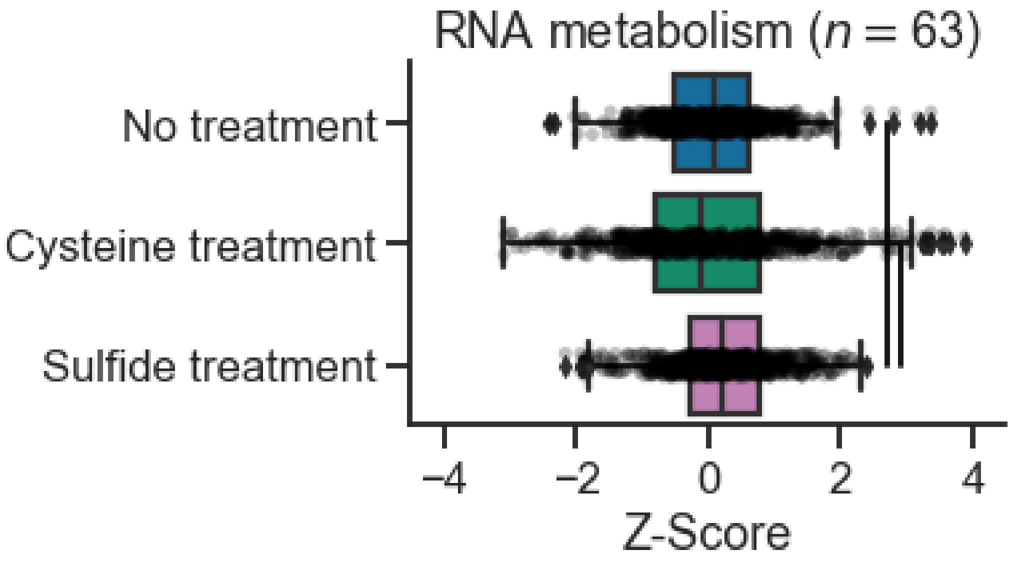
```
DEGs Cys all pathways: 23
DEGs Sulf all pathways: 0
Pathway Genes: 34
------------
DEGs Cys cys and met pathways: 19
DEGs Sulf cys and met pathways: 0
Cys and met pathway genes: 25
------------
Pathway level comparison
Cys vs Nt: 8.363129103633468e-34
Sulf vs Nt: 0.0033983789771532035
Cys vs Sulf: 9.158880058260438e-22
```



```
<Figure size 432x288 with 0 Axes>
```

```
[1013]: #SEED RNA metabolism
        genes=seed[seed.superpathway=='RNA Metabolism'].index.values
        df=pathway_degs(genes,zvals,'SEED RNA metabolism',gene_map,seed,blast_annot)
        supp_table2=pd.concat([supp_table2,df])
        pathway_boxplots(genes,'RNA metabolism',zvals,alpha=0.25)
```

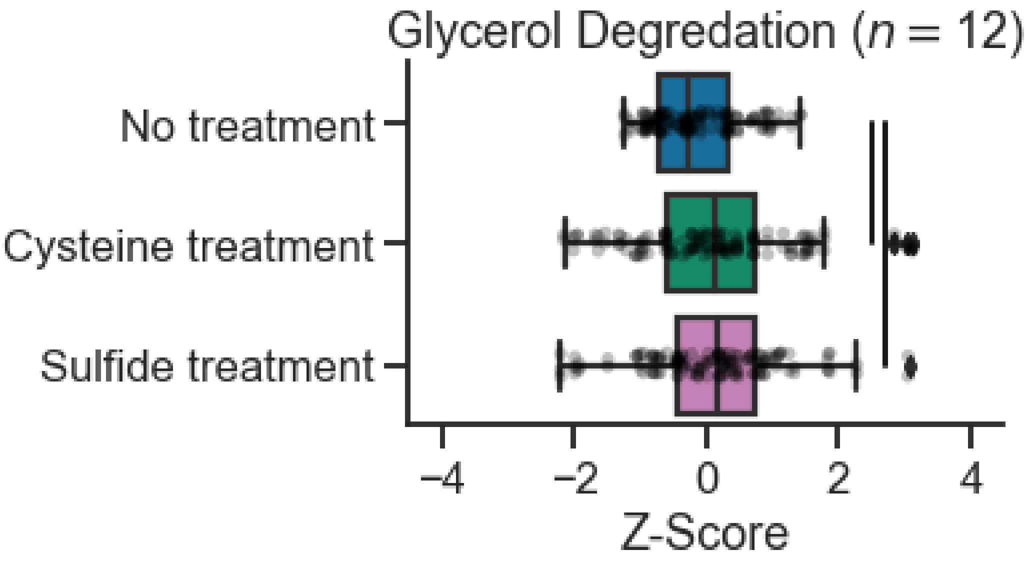
```
SEED RNA metabolism
DEGs Down Cys vs Nt: 18
DEGs Up Cys vs Nt: 14

DEGs Down Sulf vs Nt: 4
DEGs Up Sulf vs Nt: 15

DEGs Down Cys vs Sulf: 8
DEGs Up Cys vs Sulf: 10

Total Genes: 63
Cys vs Nt: 0.1454021594256219
Sulf vs Nt: 2.644204625720776e-05
Cys vs Sulf: 3.2494860441696316e-06
```



```
<Figure size 432x288 with 0 Axes>
```

```
[1014]: #MetaCyc Glycerol Degredation
        genes=path_annot[path_annot.value=='Glycerol Degredation'].ID.values
        df=pathway_degs(genes,zvals,'MetaCyc Glycerol␣
         ↪Degredation',gene_map,seed,blast_annot)
        pathway_boxplots(genes,'Glycerol Degredation',zvals,alpha=0.25,save=True)
```

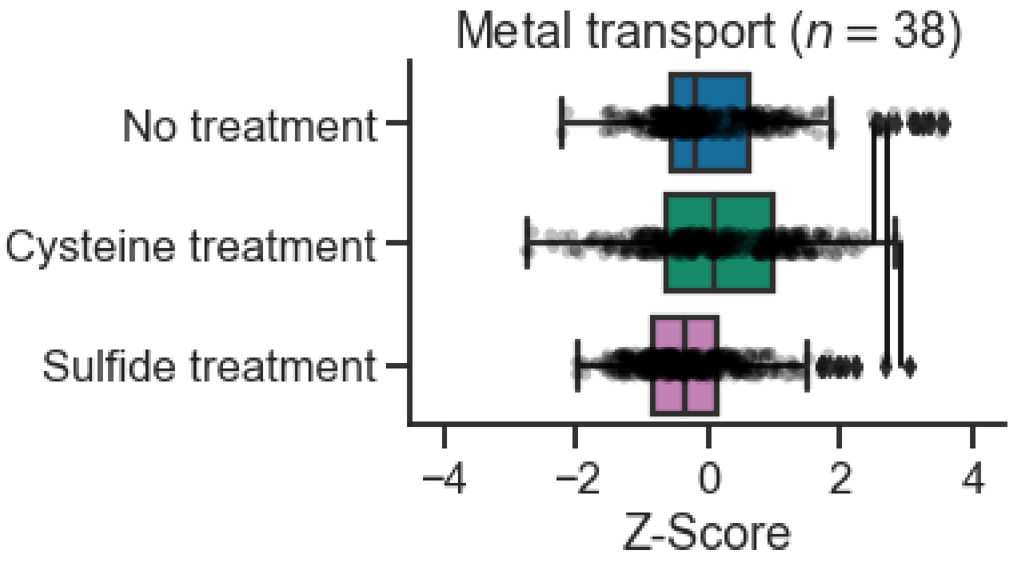
```
MetaCyc Glycerol Degredation
DEGs Down Cys vs Nt: 4
DEGs Up Cys vs Nt: 6

DEGs Down Sulf vs Nt: 2
DEGs Up Sulf vs Nt: 4

DEGs Down Cys vs Sulf: 6
DEGs Up Cys vs Sulf: 3

Total Genes: 12
Cys vs Nt: 0.004985205042491762
Sulf vs Nt: 0.019468995617144012
Cys vs Sulf: 0.49520992615783366
```



```
<Figure size 432x288 with 0 Axes>
```

```
[1015]: #Metal transporters
go_metals=['metal ion transport','copper ion transport','iron ion␣
 ↪transport','iron ion transmembrane transport']
seed_metals=['Transport of Molybdenum','Transport of Nickel and␣
 ↪Cobalt','Transport of Iron']
metal_trans=[]
res=pd.DataFrame()
for m in go_metals:
    genes=biol[biol.value==m].dropna().ID.unique()
    metal_trans+=list(genes)
    df=pathway_degs(genes,zvals,'GO '+ m,gene_map,seed,blast_annot)
    res=pd.concat([res,df])
    print('------------------')
for m in seed_metals:
    genes=seed[(seed['subpathway']==m)].index.values
    df=pathway_degs(genes,zvals,'SEED '+ m,gene_map,seed,blast_annot)
    res=pd.concat([res,df])
    metal_trans+=list(genes)
    print('------------------')
genes=pd.unique(metal_trans)
supp_table2=pd.concat([supp_table2,res])
pathway_boxplots(genes,'Metal transport',zvals,alpha=0.25,save=True)
```

```
GO metal ion transport
DEGs Down Cys vs Nt: 1
DEGs Up Cys vs Nt: 10

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 2

DEGs Down Cys vs Sulf: 2
DEGs Up Cys vs Sulf: 6

Total Genes: 14
------------------
GO copper ion transport
DEGs Down Cys vs Nt: 2
DEGs Up Cys vs Nt: 3

DEGs Down Sulf vs Nt: 3
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 2
DEGs Up Cys vs Sulf: 4

Total Genes: 6
------------------
GO iron ion transport
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 4
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 4
------------------
GO iron ion transmembrane transport
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 4
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 4
------------------
SEED Transport of Molybdenum
DEGs Down Cys vs Nt: 1
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 4
------------------
SEED Transport of Nickel and Cobalt
DEGs Down Cys vs Nt: 4
DEGs Up Cys vs Nt: 6

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 5

Total Genes: 10
------------------
SEED Transport of Iron
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 3
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 4
------------------
Cys vs Nt: 0.0435166536473679
Sulf vs Nt: 2.9257407781248126e-08
Cys vs Sulf: 8.66247281368601e-13
```



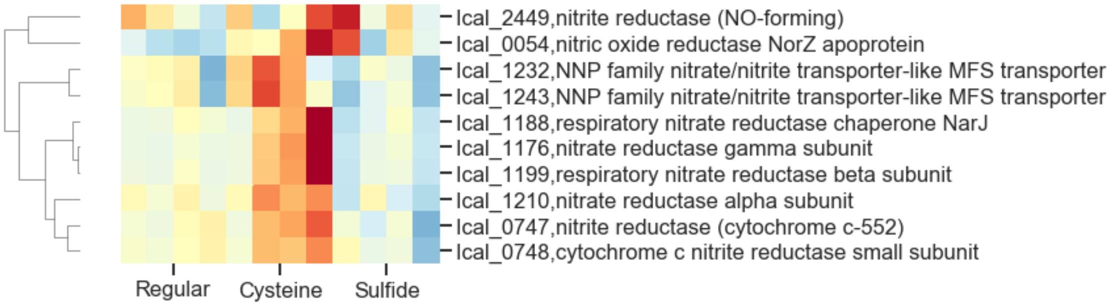
```
<Figure size 432x288 with 0 Axes>
```

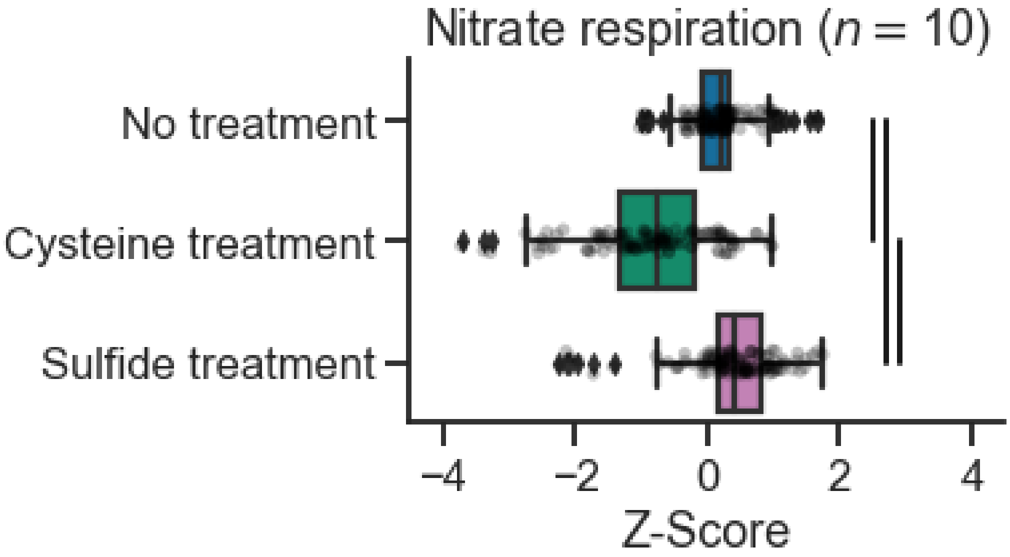
```
[1016]: #Nitrate respirtation, denitrification and DNRA
gof=pd.read_csv(r'data/Nitrate_Respiration.csv')
gof['ID']=gof['Locus tag'].map(gene_map)
genes=gof['Locus tag'].values
clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed)
df=pathway_degs(genes,zvals,'Nitrate Respiration (Manually␣
 ↪curated)',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'Nitrate respiration',zvals,alpha=0.25,save=True)
```

```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```

```
Nitrate Respiration (Manually curated)
DEGs Down Cys vs Nt: 9
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 0

Total Genes: 10
```

```
Cys vs Nt: 8.719626077715546e-21
Sulf vs Nt: 0.03913840206853242
Cys vs Sulf: 6.059142506984443e-23
```





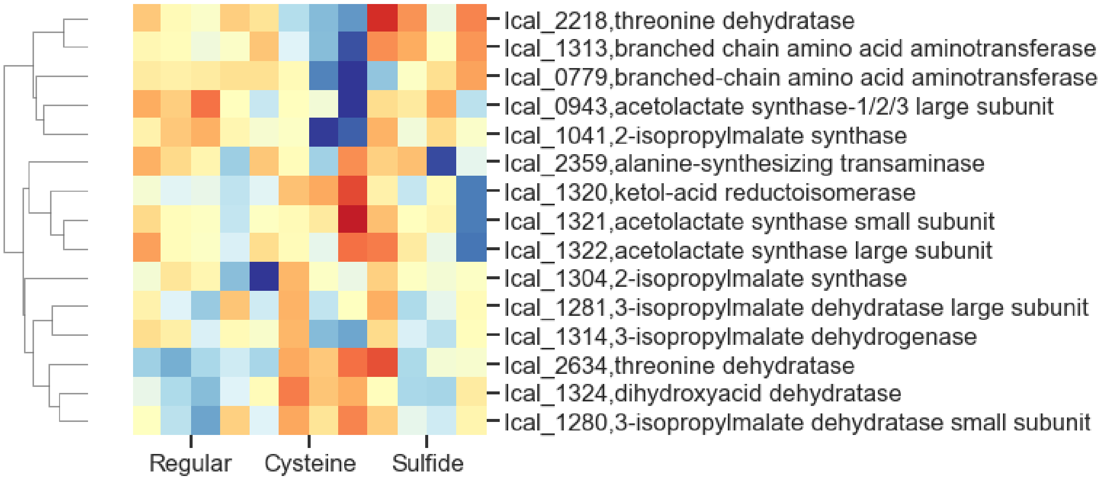
```
<Figure size 432x288 with 0 Axes>
```

[1017]:

```
#KEGG BCAA Biosynthesis
bcaa=pd.read_csv(r'data/KEGG_BCAA-Biosynthesis.csv')
bcaa['ID']=[x.split(':')[1] for x in bcaa['ID']]
bcaa
genes=[]
for ID in bcaa['ID']:
    if ID in blast_annot['KIP7.ortholog.RefSeq.old.ID'].values:
        genes.append(blast_annot[blast_annot['KIP7.ortholog.RefSeq.old.
↪ID']==ID].index.values[0])
    else:
        genes.append(np.nan)
bcaa['genes']=genes
clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed,names=bcaa.Description.
↪values)
df=pathway_degs(genes,zvals,'KEGG BCAA biosynthesis',gene_map,seed,blast_annot)
```

```
KEGG BCAA biosynthesis
DEGs Down Cys vs Nt: 4
DEGs Up Cys vs Nt: 4

DEGs Down Sulf vs Nt: 2
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 1
DEGs Up Cys vs Sulf: 4

Total Genes: 15
```

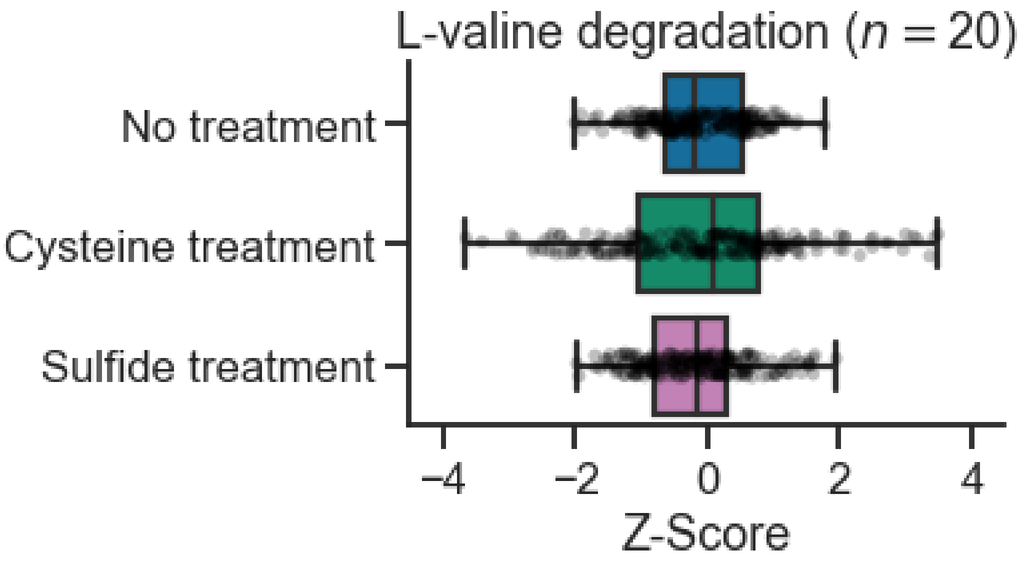
```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```



[1018]:

```
#MetaCyc Valine degredation
genes=path_annot[path_annot.value=='L-valine degradation I'].ID.unique()
df=pathway_degs(genes,zvals,'MetaCyc L-valine degradation␣
↪I',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'L-valine degradation',zvals,alpha=0.25)
```

```
MetaCyc L-valine degradation I
DEGs Down Cys vs Nt: 3
DEGs Up Cys vs Nt: 6

DEGs Down Sulf vs Nt: 5
DEGs Up Sulf vs Nt: 1

DEGs Down Cys vs Sulf: 4
DEGs Up Cys vs Sulf: 6

Total Genes: 20
Cys vs Nt: 0.36894529368439466
Sulf vs Nt: 0.42845964720448804
Cys vs Sulf: 0.15487029548099882
```



```
<Figure size 432x288 with 0 Axes>
```

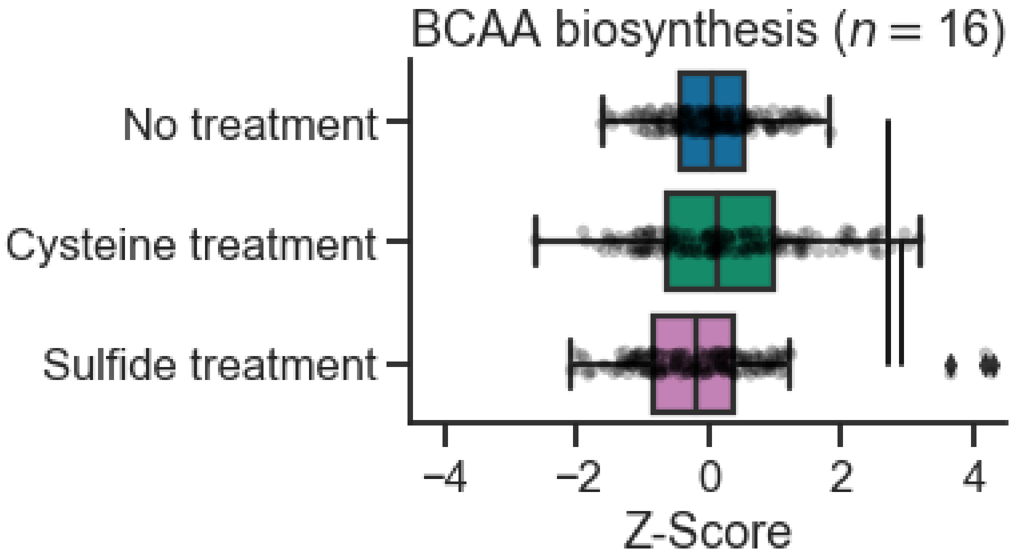
[1019]:

```
#MetaCyc BCAA
genes=paths[path_annot.value=='superpathway of branched chain amino acid␣
↪biosynthesis'].ID.unique()
```

```
df=pathway_degs(genes,zvals,'MetaCyc BCAA␣
 ↪Biosynthesis',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'BCAA biosynthesis',zvals,alpha=0.25)
```

```
MetaCyc BCAA Biosynthesis
DEGs Down Cys vs Nt: 4
DEGs Up Cys vs Nt: 5

DEGs Down Sulf vs Nt: 3
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 1
DEGs Up Cys vs Sulf: 7

Total Genes: 16
Cys vs Nt: 0.08958045161990945
Sulf vs Nt: 0.007030688999127028
Cys vs Sulf: 0.00022881654982006804
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: UserWarning:
Boolean Series key will be reindexed to match DataFrame index.
```

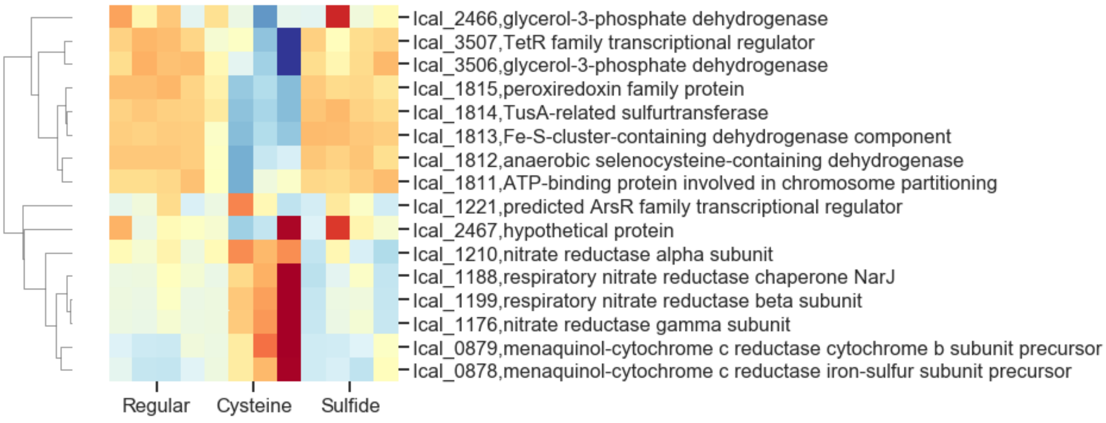


```
<Figure size 432x288 with 0 Axes>
```

```
[1020]: P= ['nitrate reduction IV (dissimilatory)',
         'nitrate reduction IX (dissimilatory)',
         'nitrate reduction V (assimilatory)',
         'nitrate reduction X (dissimilatory, periplasmic)',
           'nitrate reduction IX (dissimilatory)','glycolysis I (from glucose␣
         ↪6-phosphate)',
         'glycolysis III (from glucose)']
        i=3
        genes=path_annot[path_annot.value==P[i]].ID
        clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed)
        df=pathway_degs(genes,zvals,'MetaCyc '+P[i],gene_map,seed,blast_annot)
        supp_table2=pd.concat([supp_table2,df])
```

```
MetaCyc nitrate reduction X (dissimilatory, periplasmic)
DEGs Down Cys vs Nt: 6
DEGs Up Cys vs Nt: 8

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 1

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 8

Total Genes: 16
```

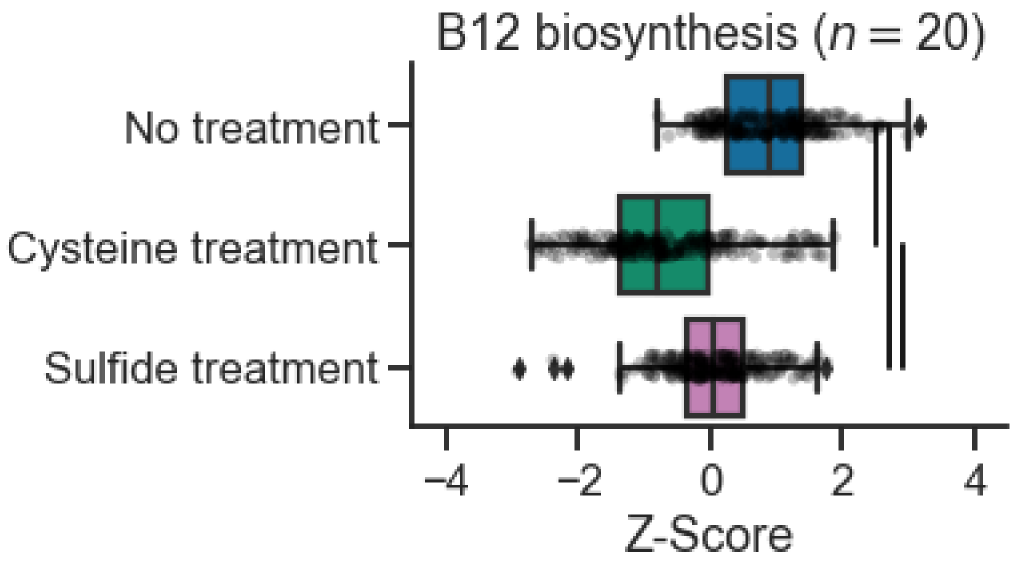
```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```



[1021]:

```
#B12 Biosynthesis
b12=pd.DataFrame()
for name in path_annot.value.dropna().unique():
    if ('cobinamide' in name.lower() or 'cobalamin' in name.lower() or␣
 ↪'dibencozide' in name.lower() or 'cob(II)yrinate' in name) and␣
 ↪'biosynthesis' in name.lower():
        print(name)
        df=path_annot[path_annot['value']==name]
        b12=pd.concat([b12,df])
print('---------------------------------------')
genes=b12.ID.unique()
df=pathway_degs(genes,zvals,'MetaCyc B12␣
 ↪Biosynthesis',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'B12 biosynthesis',zvals,alpha=0.25,save=True)
```

```
adenosylcobalamin biosynthesis from adenosylcobinamide-GDP I
2-methyladeninyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
adeninyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
5-methylbenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
5-methoxy-6-methylbenzimidazolyl adenosylcobamide biosynthesis from
adenosylcobinamide-GDP
5-methoxybenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-
GDP
5-hydroxybenzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-
GDP
benzimidazolyl adenosylcobamide biosynthesis from adenosylcobinamide-GDP
adenosylcobinamide-GDP biosynthesis from cobyrinate <i>a,c</i>-diamide
cob(II)yrinate <i>a,c</i>-diamide biosynthesis II (late cobalt incorporation)
cob(II)yrinate <i>a,c</i>-diamide biosynthesis I (early cobalt insertion)
cob(II)yrinate <i>a,c</i>-diamide biosynthesis
---------------------------------------
MetaCyc B12 Biosynthesis
DEGs Down Cys vs Nt: 17
DEGs Up Cys vs Nt: 1

DEGs Down Sulf vs Nt: 11
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 9
DEGs Up Cys vs Sulf: 2

Total Genes: 20
Cys vs Nt: 2.844672415408 15e-59
Sulf vs Nt: 4.338484609550232e-30
Cys vs Sulf: 6.867943482427032e-19
```

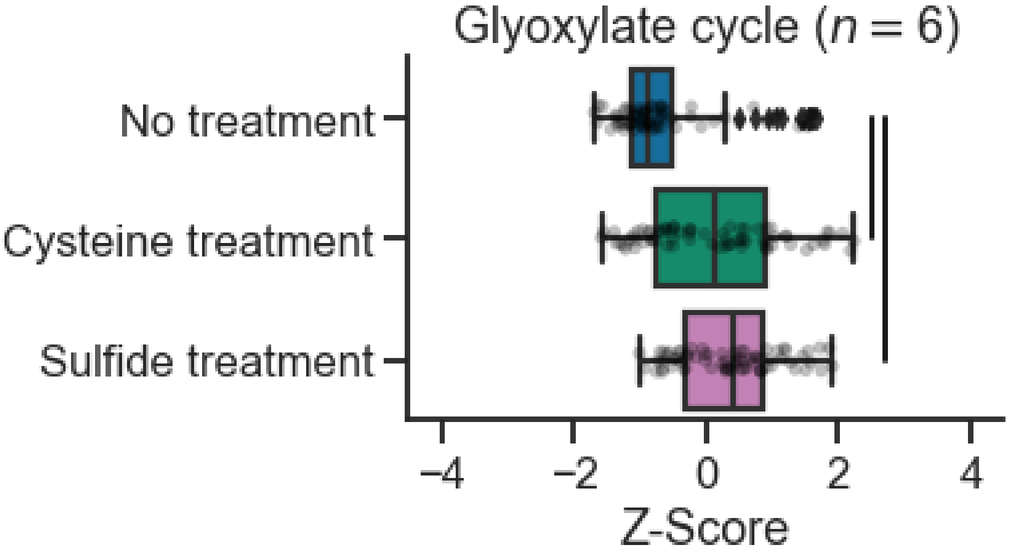


```
<Figure size 432x288 with 0 Axes>
```

```
[1022]: #Glyoxylate cycle
for name in biol.value.dropna().unique():
    if 'glyoxylate'in name.lower():
        print(name)
glyox=biol[biol.value=='glyoxylate cycle']
genes=glyox.ID.unique()
df=pathway_degs(genes,zvals,'GO Glyoxylate Cycle',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'Glyoxylate cycle',zvals,alpha=0.25,save=True)
c_genes+=list(genes)
```

```
glyoxylate cycle
GO Glyoxylate Cycle
DEGs Down Cys vs Nt: 1
DEGs Up Cys vs Nt: 4

DEGs Down Sulf vs Nt: 1
DEGs Up Sulf vs Nt: 5

DEGs Down Cys vs Sulf: 3
DEGs Up Cys vs Sulf: 0

Total Genes: 6
Cys vs Nt: 6.715538779514185e-05
Sulf vs Nt: 8.477520578607683e-10
```

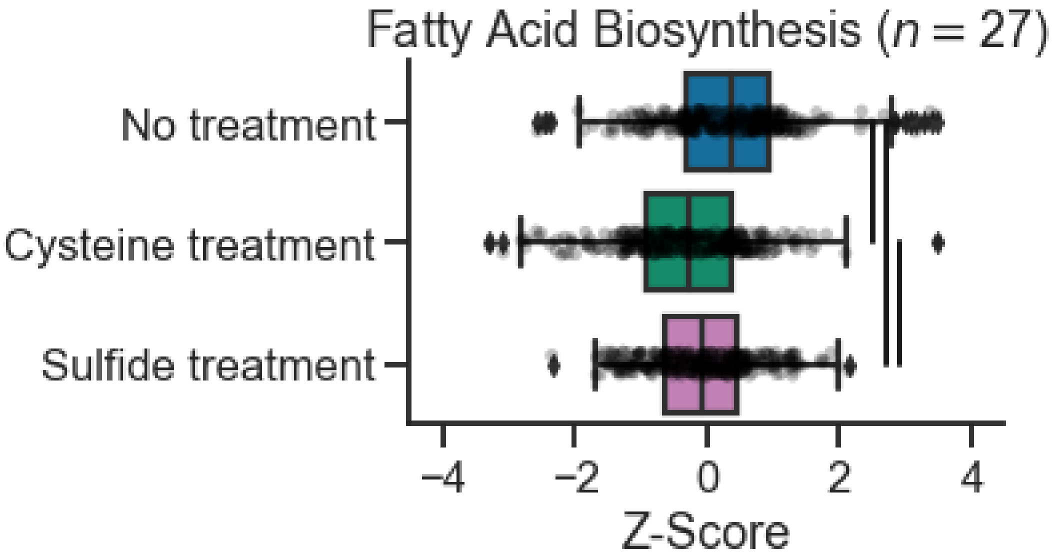
```
Cys vs Sulf: 0.09431957044753418
```



```
<Figure size 432x288 with 0 Axes>
```

[1023]:
```
#Fatty acid biosynthesis
fatty=pd.DataFrame()
for name in path_annot.value.dropna().unique():
    if 'fatty acid' in name.lower() and 'biosynthesis' in name.lower():
        print(name)
        df=path_annot[path_annot['value']==name]
        fatty=pd.concat([fatty,df])
print('-------')
genes=fatty.ID.unique()
df=pathway_degs(genes,zvals,'MetaCyc Fatty Acid␣
 ↪Biosynthesis',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'Fatty Acid Biosynthesis',zvals,alpha=0.25,save=True)
```

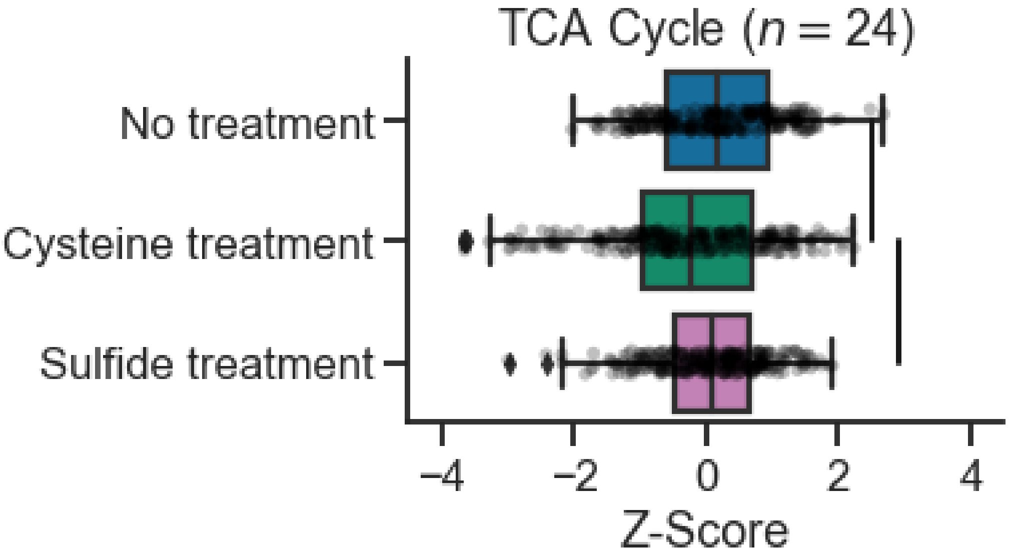
```
cyclopropane fatty acid (CFA) biosynthesis
Fatty Acid and Lipid Biosynthesis
Unsaturated Fatty Acid Biosynthesis
Cyclopropane Fatty Acid Biosynthesis
Fatty Acid Biosynthesis
-------
MetaCyc Fatty Acid Biosynthesis
DEGs Down Cys vs Nt: 13
DEGs Up Cys vs Nt: 2
DEGs Down Sulf vs Nt: 13
DEGs Up Sulf vs Nt: 2

DEGs Down Cys vs Sulf: 6
DEGs Up Cys vs Sulf: 2

Total Genes: 27
Cys vs Nt: 1.1526913912508817e-15
Sulf vs Nt: 2.0044994341240444e-08
Cys vs Sulf: 0.001264571007085756
```



```
<Figure size 432x288 with 0 Axes>
```

```
[1024]: #MetaCyc TCA cycle
tca=pd.DataFrame()
for name in path_annot.value.dropna().unique():
    if 'tca' in name.lower() in name.lower():
        print(name)
        df=path_annot[path_annot['value']==name]
        tca=pd.concat([tca,df])
print('--------')
genes=tca.ID.unique()
df=pathway_degs(genes,zvals,'MetaCyc TCA Cycle',gene_map,seed,blast_annot)
pathway_boxplots(genes,'TCA Cycle',zvals,alpha=0.25,save=True)
supp_table2=pd.concat([supp_table2,df])
c_genes+=list(genes)
```

```
TCA cycle V (2-oxoglutarate:ferredoxin oxidoreductase)
TCA cycle
-------
MetaCyc TCA Cycle
DEGs Down Cys vs Nt: 10
DEGs Up Cys vs Nt: 6

DEGs Down Sulf vs Nt: 9
DEGs Up Sulf vs Nt: 6

DEGs Down Cys vs Sulf: 8
DEGs Up Cys vs Sulf: 5

Total Genes: 24
Cys vs Nt: 2.7909105560718356e-06
Sulf vs Nt: 0.11290979718570444
Cys vs Sulf: 0.00039854553473864327
```



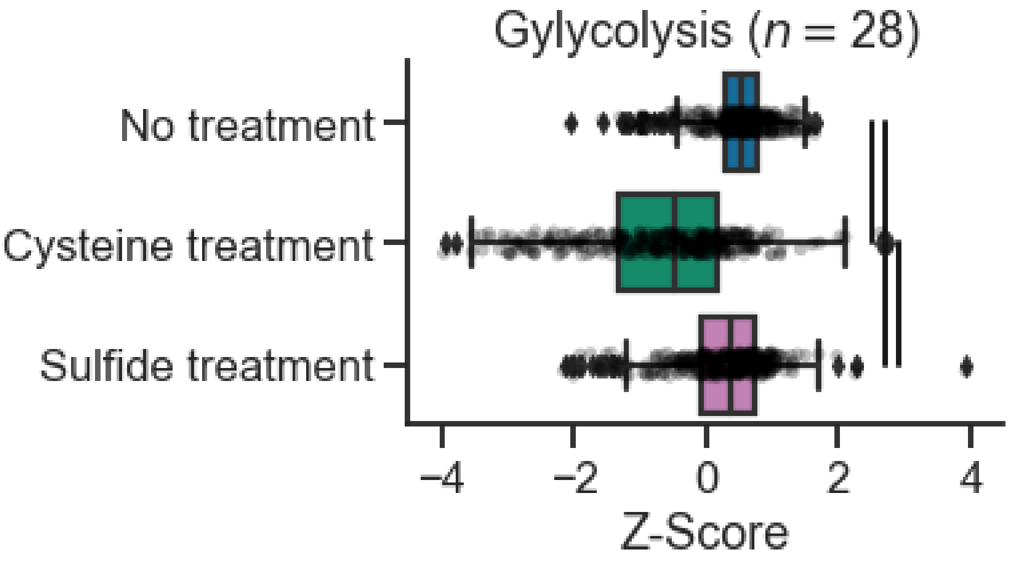
```
<Figure size 432x288 with 0 Axes>
```

[1025]:

```
#MetaCyc glycolysis
glyc=pd.DataFrame()
for name in path_annot.value.dropna().unique():
    if 'glycolysis' in name.lower() in name.lower():
        print(name)
        df=path_annot[path_annot['value']==name]
        glyc=pd.concat([glyc,df])
print('-------')
```

```
genes=glyc.ID.unique()
df=pathway_degs(genes,zvals,'MetaCyc Gylycolysis',gene_map,seed,blast_annot)
pathway_boxplots(genes,'Gylycolysis',zvals,alpha=0.25,save=True)
supp_table2=pd.concat([supp_table2,df])
```

```
glycolysis I (from glucose 6-phosphate)
glycolysis III (from glucose)
Glycolysis
-------
MetaCyc Gylycolysis
DEGs Down Cys vs Nt: 23
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 6
DEGs Up Sulf vs Nt: 2

DEGs Down Cys vs Sulf: 5
DEGs Up Cys vs Sulf: 3

Total Genes: 28
Cys vs Nt: 4.236998976601631e-46
Sulf vs Nt: 0.0002351624817312966
Cys vs Sulf: 4.203553624636403e-32
```

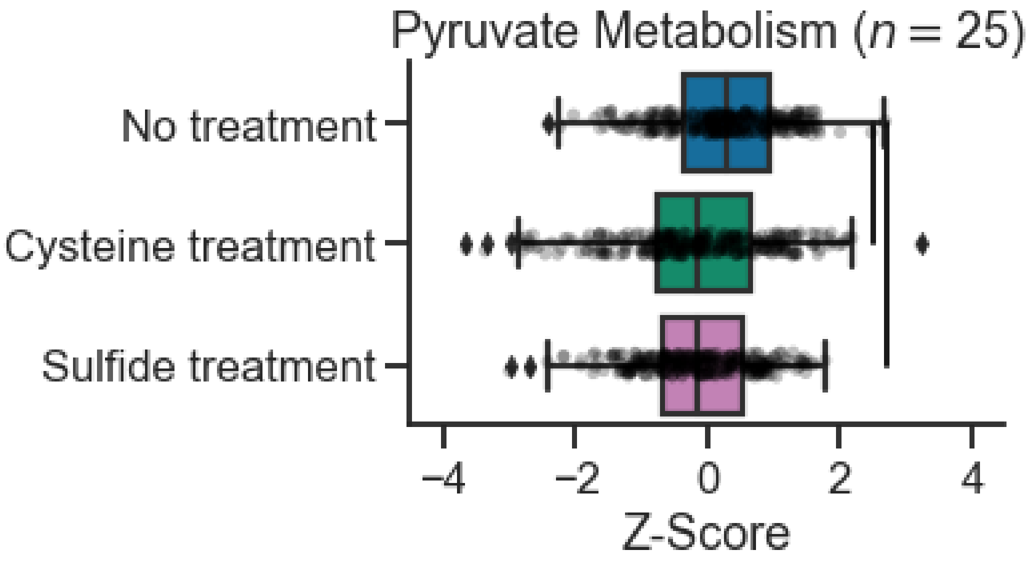


```
<Figure size 432x288 with 0 Axes>
```

[1026]:

```
#MetaCyc pyruvate
pyv=pd.DataFrame()
for name in path_annot.value.dropna().unique():
    if 'pyruvate' in name.lower() and ('ferm' in name.lower() or 'decarb' in␣
 ↪name.lower()):
        print(name)
        df=path_annot[path_annot['value']==name]
        pyv=pd.concat([pyv,df])
print('-------')
genes=pyv.ID.unique()
df=pathway_degs(genes,zvals,'MetaCyc Pyruvate␣
 ↪Metabolism',gene_map,seed,blast_annot)
pathway_boxplots(genes,'Pyruvate Metabolism',zvals,alpha=0.25,save=True)
supp_table2=pd.concat([supp_table2,df])
```

```
pyruvate decarboxylation to acetyl CoA
pyruvate fermentation to acetate IV
Pyruvate Fermentation to Ethanol
Pyruvate Fermentation to Acetate
-------
MetaCyc Pyruvate Metabolism
DEGs Down Cys vs Nt: 12
DEGs Up Cys vs Nt: 5

DEGs Down Sulf vs Nt: 10
DEGs Up Sulf vs Nt: 3

DEGs Down Cys vs Sulf: 6
DEGs Up Cys vs Sulf: 10

Total Genes: 25
Cys vs Nt: 7.639117353236542e-07
Sulf vs Nt: 2.3374070254521258e-08
Cys vs Sulf: 0.85769050578437
```



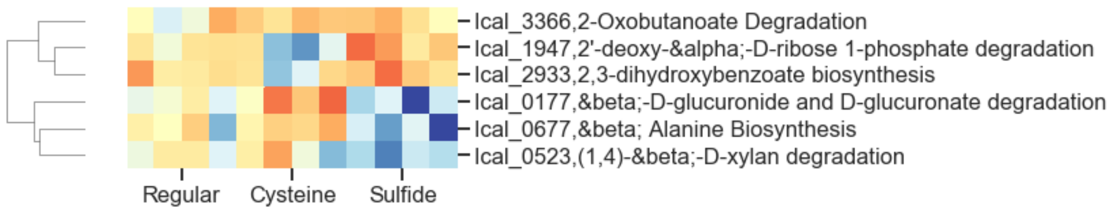
```
<Figure size 432x288 with 0 Axes>
```

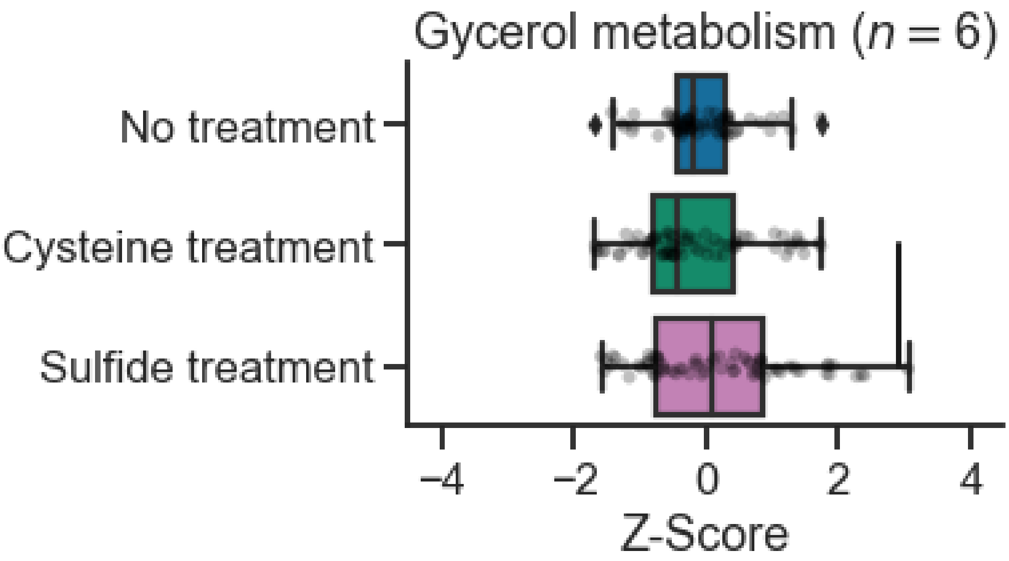
[1027]:
```
#SEED glycerol transport and utilization
cond=['Regular','Cysteine-inhibition','Sulfide-inhibition']
genes=seed[seed.subpathway=='Glycerol and Glycerol-3-phosphate Uptake and␣
 ↪Utilization'].index
genes=list(genes)
genes.append('Ga0242637_11703')

clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed,names=names)
df=pathway_degs(genes,zvals,'SEED Glycerol Transport',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'Gycerol metabolism',zvals,alpha=0.25)
```

```
SEED Glycerol Transport
DEGs Down Cys vs Nt: 3
DEGs Up Cys vs Nt: 2

DEGs Down Sulf vs Nt: 1
DEGs Up Sulf vs Nt: 3

DEGs Down Cys vs Sulf: 3
DEGs Up Cys vs Sulf: 2

Total Genes: 6
Cys vs Nt: 0.46790329262105734
Sulf vs Nt: 0.06694833058843964
```

```
Cys vs Sulf: 0.029298327818286066
```

```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```





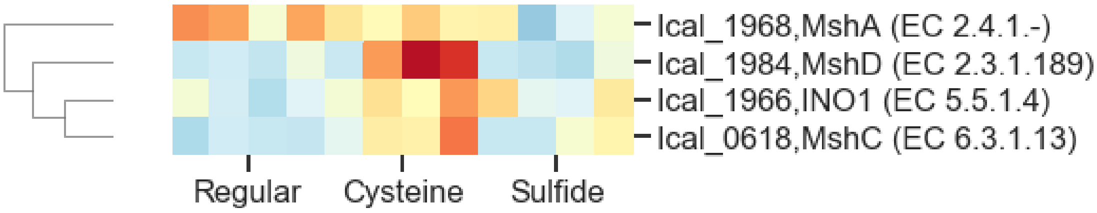
```
<Figure size 432x288 with 0 Axes>
```

```
[1028]: #GO mycothiol biosynthesis
        cond=['Regular','Cysteine-inhibition','Sulfide-inhibition']
        genes=go_terms[go_terms.value=='mycothiol biosynthetic process'].ID.values
        names=[]
        for i in myco_genes:
            if i in seed.index.values:
                names.append(str(seed.loc[i,'function']))
            else:
```

```
        names.append('mycothiol synthase activity')
names = ['INO1 (EC 5.5.1.4)','MshA (EC 2.4.1.-)','MshC (EC 6.3.1.13)','MshD (EC␣
 ↪2.3.1.189)']
genes=[genes[-1],genes[3],genes[0],genes[1]]
clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed,names=names)
df=pathway_degs(genes,zvals,'GO Mycothiol␣
 ↪Biosynthesis',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
```

```
GO Mycothiol Biosynthesis
DEGs Down Cys vs Nt: 3
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 2
DEGs Up Sulf vs Nt: 1

DEGs Down Cys vs Sulf: 3
DEGs Up Cys vs Sulf: 0

Total Genes: 4
```

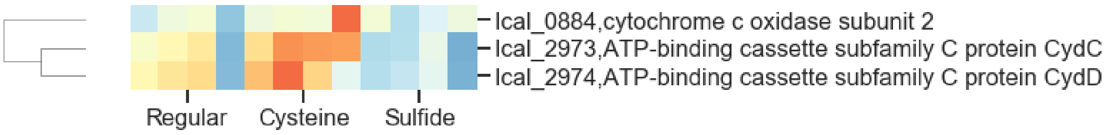
```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The bottom and top margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```

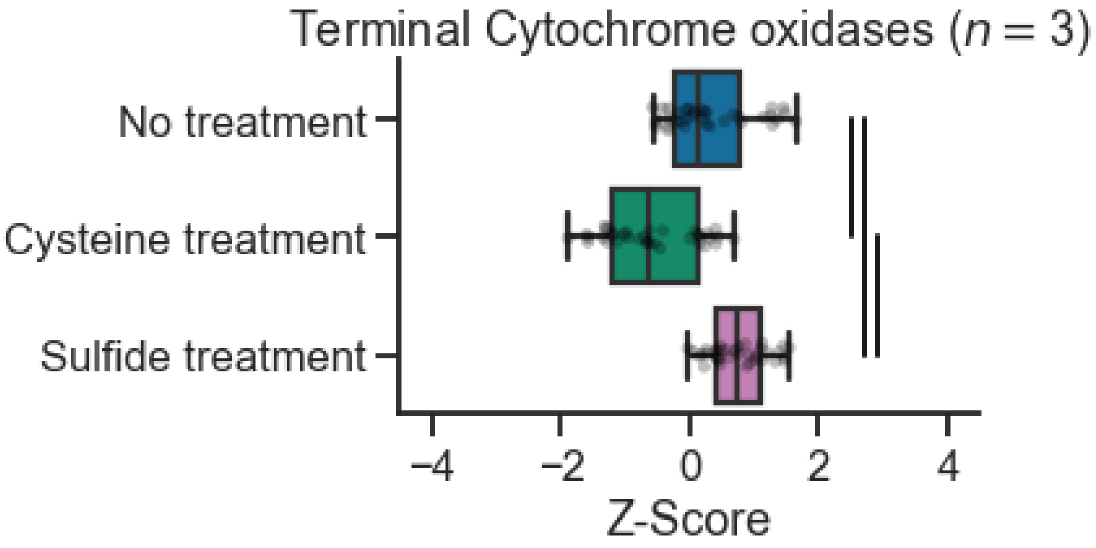


[1036]:
```
#Terminal cytochrome oxidases
genes=seed[(seed['subpathway']=='Terminal cytochrome C␣
 ↪oxidases')|(seed['subpathway']=='Terminal cytochrome oxidases')].index.
 ↪unique().values
clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed)
df=pathway_degs(genes,zvals,'SEED Terminal Cytochrome␣
 ↪Oxidases',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'Terminal Cytochrome oxidases',zvals,alpha=0.
 ↪25,save=True)
```

```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```

```
SEED Terminal Cytochrome Oxidases
DEGs Down Cys vs Nt: 3
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 2

DEGs Down Cys vs Sulf: 2
DEGs Up Cys vs Sulf: 0

Total Genes: 3
Cys vs Nt: 6.152781792627683e-07
Sulf vs Nt: 0.0014246633547885738
Cys vs Sulf: 1.2458354906846305e-13
```



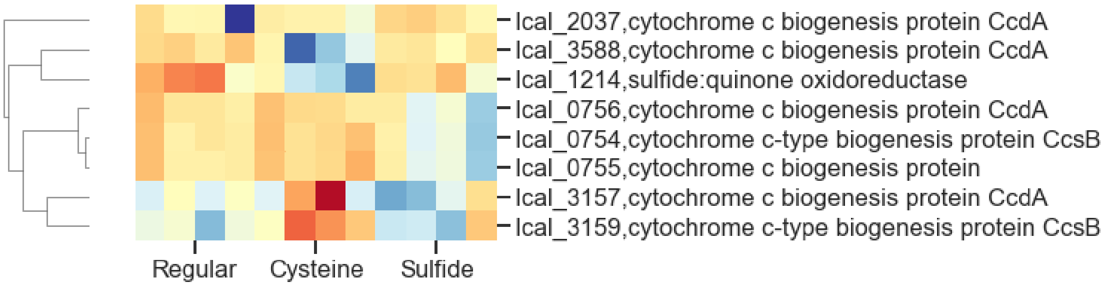


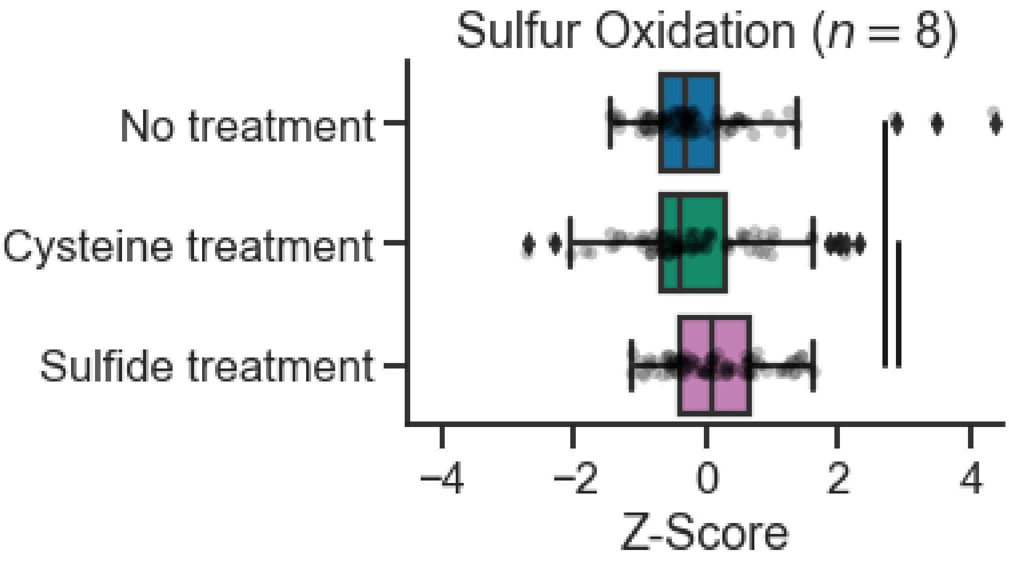
```
<Figure size 432x288 with 0 Axes>
```

```
[1038]: #Terminal cytochrome oxidases, Sulfur Oxidation
        genes=seed[(seed['subpathway']=='YedY-YedZ␣
         ↪cluster')|(seed['subpathway']=='Sulfur oxidation')].index.unique().values
        clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed)
        df=pathway_degs(genes,zvals,'SEED Sulfur Oxidation',gene_map,seed,blast_annot)
        supp_table2=pd.concat([supp_table2,df])
        pathway_boxplots(genes,'Sulfur Oxidation',zvals,alpha=0.25,save=True)
```

```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```

```
SEED Sulfur Oxidation
DEGs Down Cys vs Nt: 1
DEGs Up Cys vs Nt: 2

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 4

DEGs Down Cys vs Sulf: 3
DEGs Up Cys vs Sulf: 3

Total Genes: 8
Cys vs Nt: 0.7831340688168883
Sulf vs Nt: 0.00965791960331437
Cys vs Sulf: 0.0047907859545647915
```





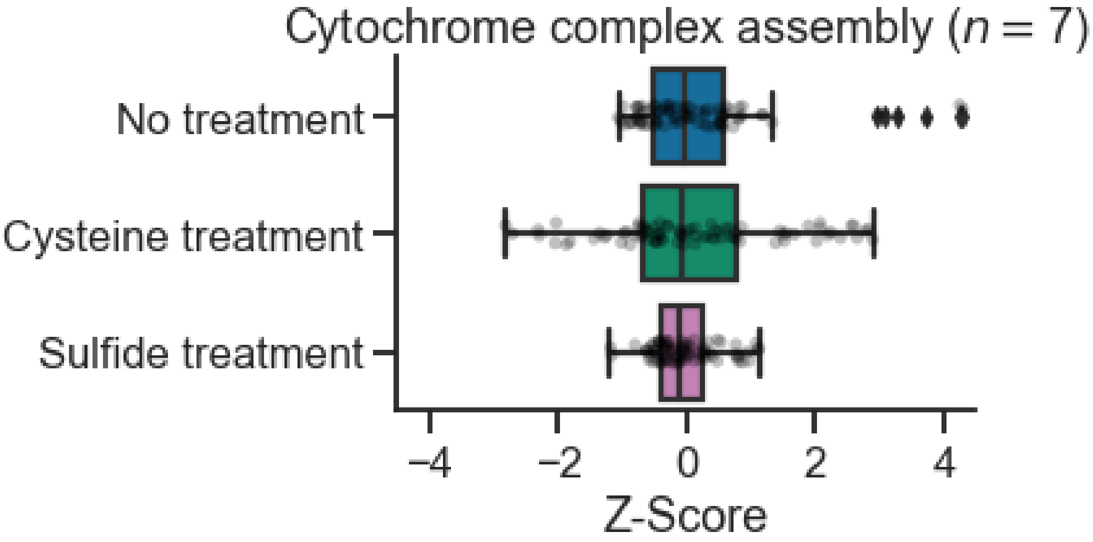
```
<Figure size 432x288 with 0 Axes>
```

[1030]:

```
#Cytochrome assembly
seed_cyt=seed[(seed['subpathway']=='YedY-YedZ␣
↪cluster')|(seed['subpathway']=='Terminal cytochrome C␣
↪oxidases')|(seed['subpathway']=='Terminal cytochrome␣
↪oxidases')|(seed['subpathway']=='Sulfur oxidation')].index.unique().values
genes=biol[biol.value=='cytochrome complex assembly'].ID.values
genes=list(set(genes)-set(seed_cyt))
df=pathway_degs(genes,zvals,'GO Cytochrome Complex␣
↪Assembly',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'Cytochrome complex assembly',zvals,alpha=0.25)
```

```
GO Cytochrome Complex Assembly
DEGs Down Cys vs Nt: 2
DEGs Up Cys vs Nt: 3

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 1

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 7
Cys vs Nt: 0.7463114708158275
Sulf vs Nt: 0.08648556190090918
```

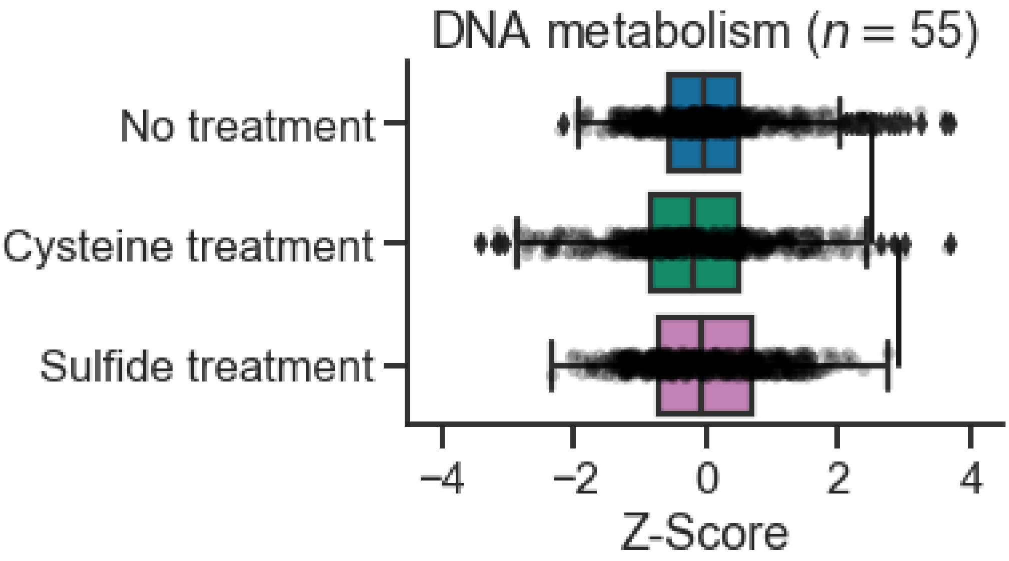
```
Cys vs Sulf: 0.2839905707804076
```



```
<Figure size 432x288 with 0 Axes>
```

[1031]:
```
#SEED DNA Metabolism
genes=seed[seed['superpathway']=='DNA Metabolism'].index.unique().values
df=pathway_degs(genes,zvals,'SEED DNA Metabolism',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
pathway_boxplots(genes,'DNA metabolism',zvals,alpha=0.25)
```

```
SEED DNA Metabolism
DEGs Down Cys vs Nt: 15
DEGs Up Cys vs Nt: 9

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 7

Total Genes: 55
Cys vs Nt: 0.00017508251025303694
Sulf vs Nt: 0.35914338191354234
Cys vs Sulf: 0.0037077824697788753
```



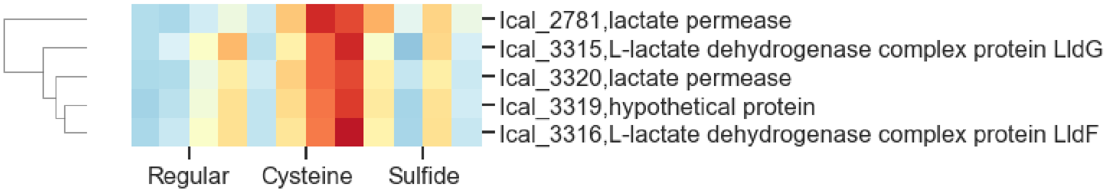
```
<Figure size 432x288 with 0 Axes>
```

[1032]:

```
#Lactate transport
g=biol[(biol.value=='lactate transmembrane transport')|(biol.value=='lactate␣
 ↪transport')].ID.unique()
genes=seed.loc[g].iloc[:5,:].index.values
clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed)
df=pathway_degs(genes,zvals,'SEED Lactate Transport',gene_map,seed,blast_annot)
supp_table2=pd.concat([supp_table2,df])
```

```
SEED Lactate Transport
DEGs Down Cys vs Nt: 5
DEGs Up Cys vs Nt: 0

DEGs Down Sulf vs Nt: 1
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 1
DEGs Up Cys vs Sulf: 0

Total Genes: 5
```

```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```

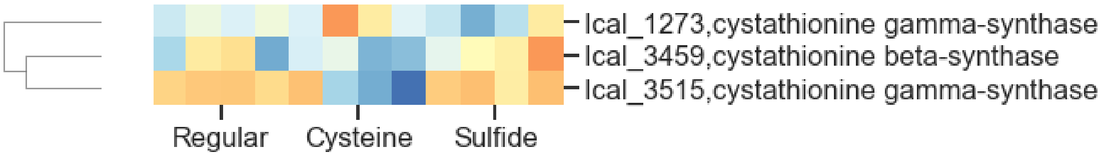


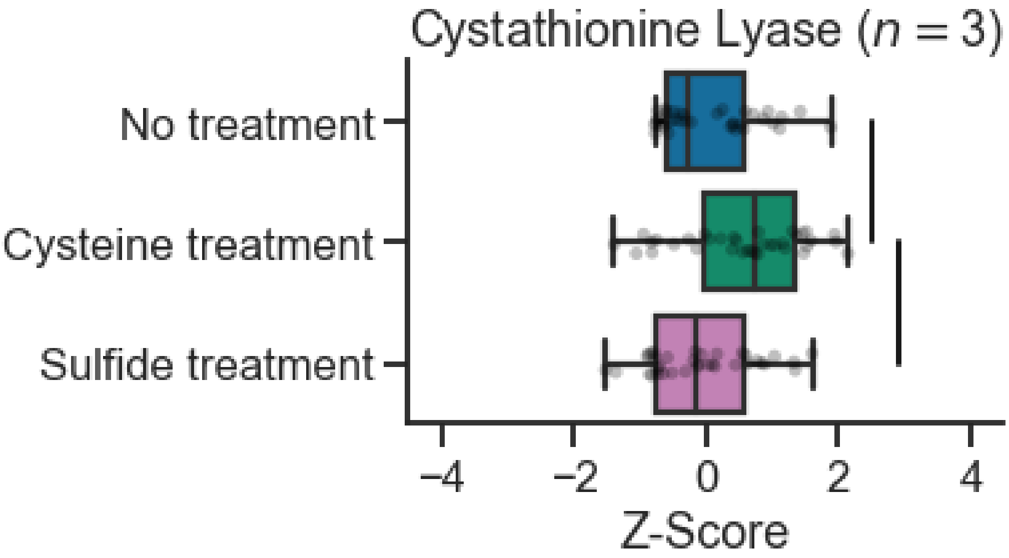
[1033]:

```
genes=[]
for i in seed['function'].dropna().index:
    name=seed.loc[i,'function']
    if type(name) != type(''):
        name=','.join(name.values)
    if 'cystathionine' in name.lower() or '4.4.1.1' in name or 'gamma-lyase' in␣
 ↪name.lower():
        print(gene_map[i],name)
        genes.append(i)

clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed)
df=pathway_degs(genes,zvals,'Cystathionine Lyase (manually␣
 ↪curated)',gene_map,seed,blast_annot)
pathway_boxplots(genes,'Cystathionine Lyase',zvals,alpha=0.25)
```

```
Ical_3459 Cystathionine beta-synthase (EC 4.2.1.22)
Ical_3515 Cystathionine gamma-lyase (EC 4.4.1.1)
Ical_1273 Cystathionine gamma-synthase (EC 2.5.1.48)
Cystathionine Lyase (manually curated)
DEGs Down Cys vs Nt: 0
DEGs Up Cys vs Nt: 1

DEGs Down Sulf vs Nt: 0
DEGs Up Sulf vs Nt: 0

DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 2

Total Genes: 3
Cys vs Nt: 0.01268301858048739
Sulf vs Nt: 0.422883102552138
Cys vs Sulf: 0.002031257753877047
```

```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The bottom and top margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```



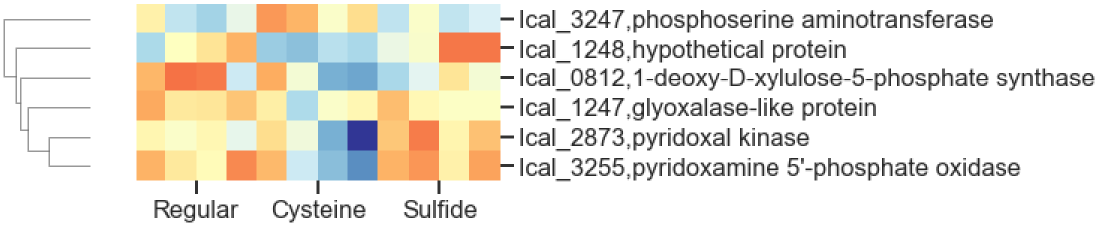


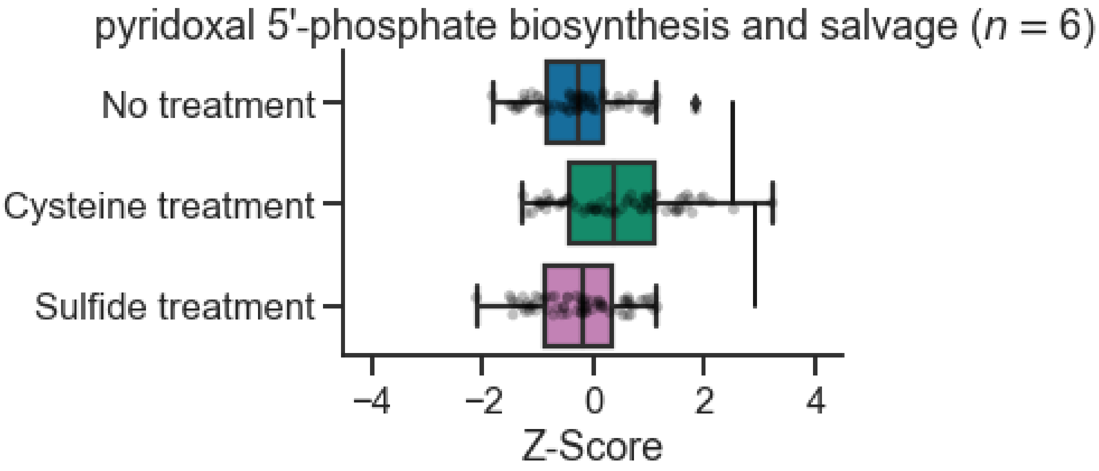
```
<Figure size 432x288 with 0 Axes>
```

[1034]:
```
#pyridoxal 5'-phosphate biosynthesis and salvage
genes=path_annot[path_annot['value']=="superpathway of pyridoxal 5'-phosphate␣
 ↪biosynthesis and salvage"].ID.values
clust_heatmap(genes,avgzvals,gene_map,blast_annot,seed)
df=pathway_degs(genes,zvals,"MetaCyc pyridoxal 5'-phosphate biosynthesis and␣
 ↪salvage",gene_map,seed,blast_annot)
pathway_boxplots(genes,"pyridoxal 5'-phosphate biosynthesis and␣
 ↪salvage",zvals,alpha=0.25)
supp_table2=pd.concat([supp_table2,df])
```

```
MetaCyc pyridoxal 5'-phosphate biosynthesis and salvage
DEGs Down Cys vs Nt: 1
DEGs Up Cys vs Nt: 5

DEGs Down Sulf vs Nt: 1
DEGs Up Sulf vs Nt: 1
DEGs Down Cys vs Sulf: 0
DEGs Up Cys vs Sulf: 3

Total Genes: 6
Cys vs Nt: 6.9017706652427795e-06
Sulf vs Nt: 0.9532872197886644
Cys vs Sulf: 9.955021571449115e-06
```

```
//anaconda3/lib/python3.7/site-packages/seaborn/matrix.py:1204: UserWarning:
Tight layout not applied. The left and right margins cannot be made large enough
to accommodate all axes decorations.
  self.fig.tight_layout(**tight_params)
```





```
<Figure size 432x288 with 0 Axes>
```

[1035]:
```
#Save tables generated
```

```
supp_tabled.
 ↪drop_duplicates()[['name','enrichement','clust','clust_n','null_n','percent','pval','avg_zs
 ↪to_csv('Supplementary_table1.csv',index=False)
supp_table2.drop_duplicates()[['Gene ID','pathway','description','Avg Z-score␣
 ↪Nt','Avg Z-score Cys','Avg Z-score Sulf','Cys-vs-Nt p-value','Sulf-vs-Nt␣
 ↪p-value']].to_csv('Supplementary_table2.csv',index=False)
```

13 Congrats, you made it to the end!

- Hopefully everyting made sense
- Time to celebrate your achievements and try to make sense of all this data!