RNAi_analysis-with_highlighting-final-Copy1

August 3, 2023

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import glob
import seaborn as sns
```

[4]:
```
path = r'D:\Data\RNAi_screen' # use your path with annotation list
files = glob.glob(path + "/*/*.csv")
```

[5]:
```
#Create dictionary with strain names and well numbers
file="D:/Data/RNAi_screen/All_RNAi_thawed.xlsx"
df_strain = pd.read_excel(file, sheet_name="Allaccumulated(2)")
df_dict = pd.DataFrame(columns=['Plate_well','Genotype'])
df_dict['Plate_well'] = df_strain['Plate'].str.slice(6)+"-"+df_strain['Well']
df_dict['Genotype'] = df_strain['Gene']
Dict = pd.Series(df_dict.Genotype.values,index=df_dict.Plate_well).to_dict()
```

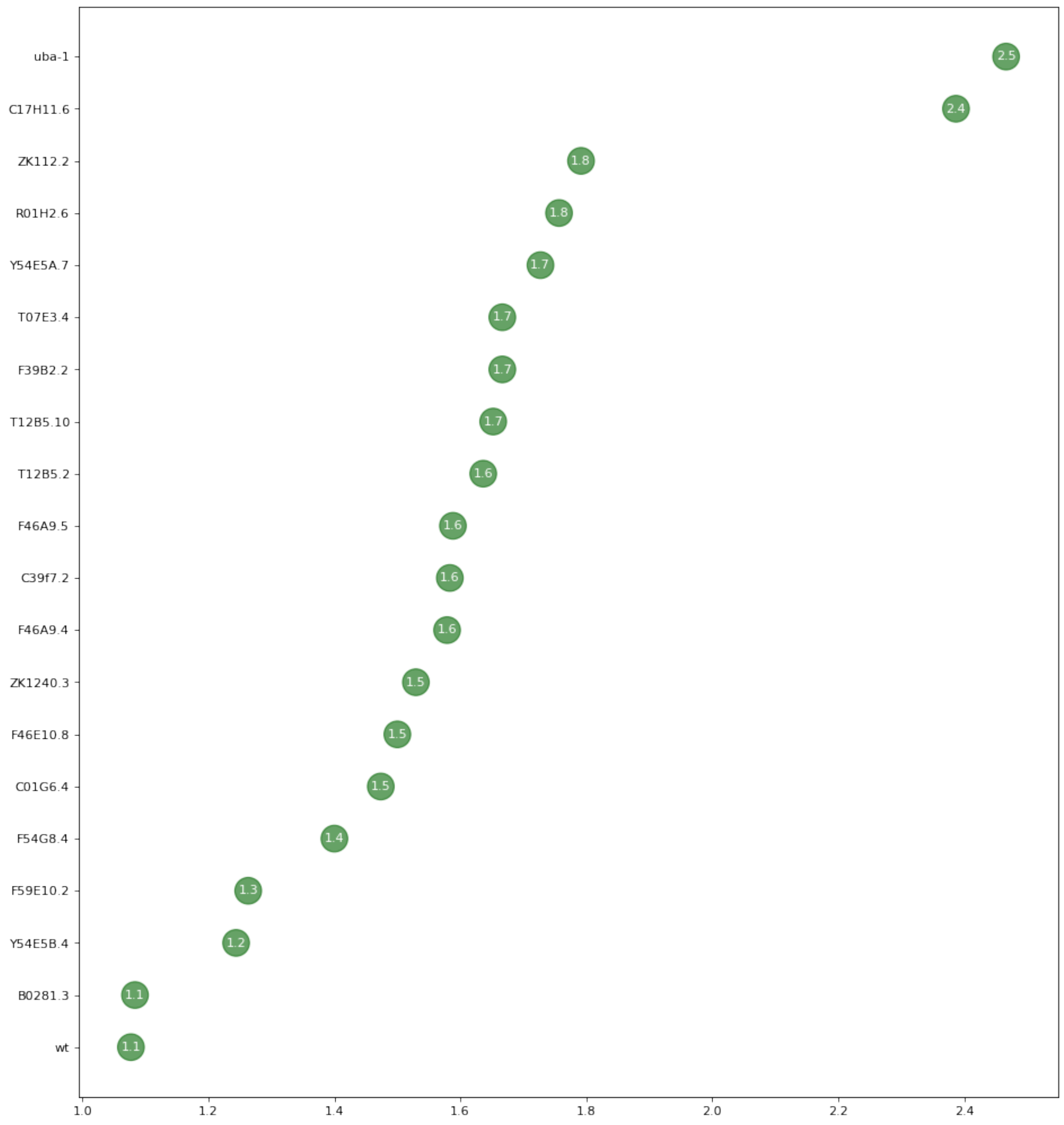
[6]:
```
#Read file
df = None
for i, f in enumerate (files):
    if i == 0:
        df = pd.read_csv(f, sep=',', error_bad_lines=False)
        df['Strain_name'] = f.split('\\')[-1][:-4]
    else:
        tmp = pd.read_csv(f, sep=',', error_bad_lines=False)
        #print(f)
        tmp['Strain_name'] = f.split('\\')[-1][:-4]
        df = df.append(tmp)
df.replace({'Strain_name':Dict}, inplace=True) #Change plate and well IDs to␣
 ↪genotypes
#print(df.to_string())
#pd.set_option('display.max_columns', 30)
df.head()
```

```
b'Skipping line 3: expected 12 fields, saw 23\n'
```

[6]:
```
  PLM cell body PLM process PLM branch PLM synapse PLM distal tip  \
0           med     low-med        med         med            med
1           med     low-med    low-med         med            med
```

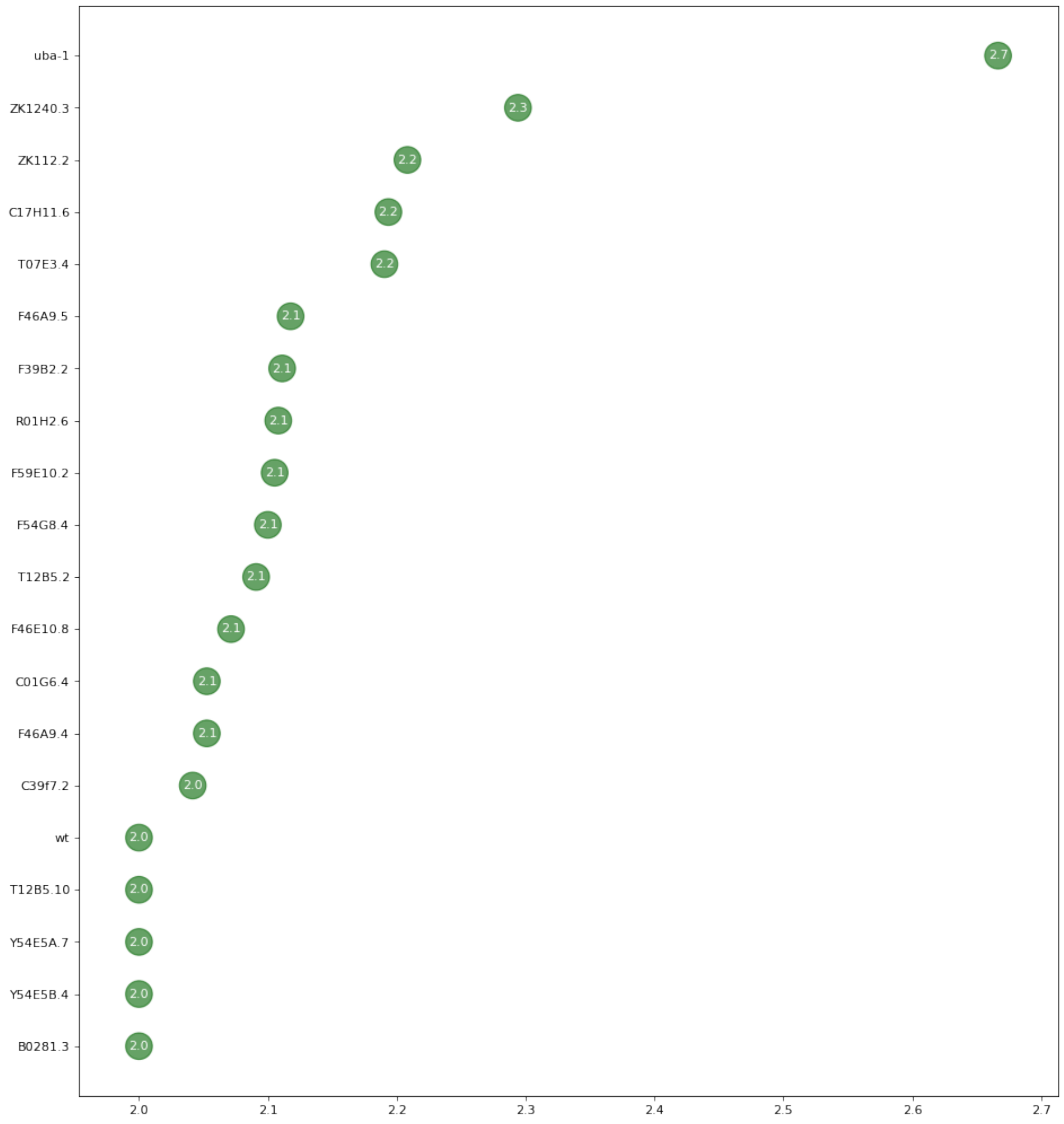
```
2           med     low-med        med    med-high           high
3           med     low-med    low-med         med       med-high
4           med     low-med        med    med-high        low-med

  ALM cell body ALM process ALM branch ALM synapse ALM distal tip  \
0           med         med        med         med        low-med
1           med     low-med    low-med         med        low-med
2           low         low        low         low        low-med
3           med     low-med    low-med     low-med        low-med
4           med         med        med         med        low-med

                    Datetime Other comments Strain_name
0  2019-06-04 21:49:41.064573              0     C28G1.1
1  2019-06-04 21:50:07.990220              0     C28G1.1
2  2019-06-04 21:50:49.423893              0     C28G1.1
3  2019-06-04 21:51:21.887550              0     C28G1.1
4  2019-06-04 21:51:49.327998              0     C28G1.1
```

[5]:
```
df2 = df.replace(["low","low-med","med","med-high","high"], [int(0.0),int(1.
↪0),int(2.0),int(3.0),int(4.0)])
dfmean = df2.groupby('Strain_name', as_index=False).mean()

x = dfmean.loc[:, ['PLM distal tip']]
dfmean['Intensity_z'] = (x)# - x.mean())/x.std()
dfmean['colors'] = ['red' if x < 0 else 'darkgreen' for x in␣
↪dfmean['Intensity_z']]
dfmean.sort_values('Intensity_z', inplace=True)
dfmean.reset_index(inplace=True)
#dfmean.set_index("Strain_name",drop=True,inplace=True)
plt.figure(figsize=(14,16), dpi= 80)
plt.scatter(dfmean.Intensity_z, dfmean.index, s=450, alpha=.6, color=dfmean.
↪colors)
for x, y, tex in zip(dfmean.Intensity_z, dfmean.index, dfmean.Intensity_z):
    t = plt.text(x, y, round(tex, 1), horizontalalignment='center',
                 verticalalignment='center', fontdict={'color':'white'})

plt.yticks(dfmean.index, dfmean.Strain_name)
#dfmean.plot.bar(y="PLM distal tip")
#plt.tight_layout()
plt.show()
```

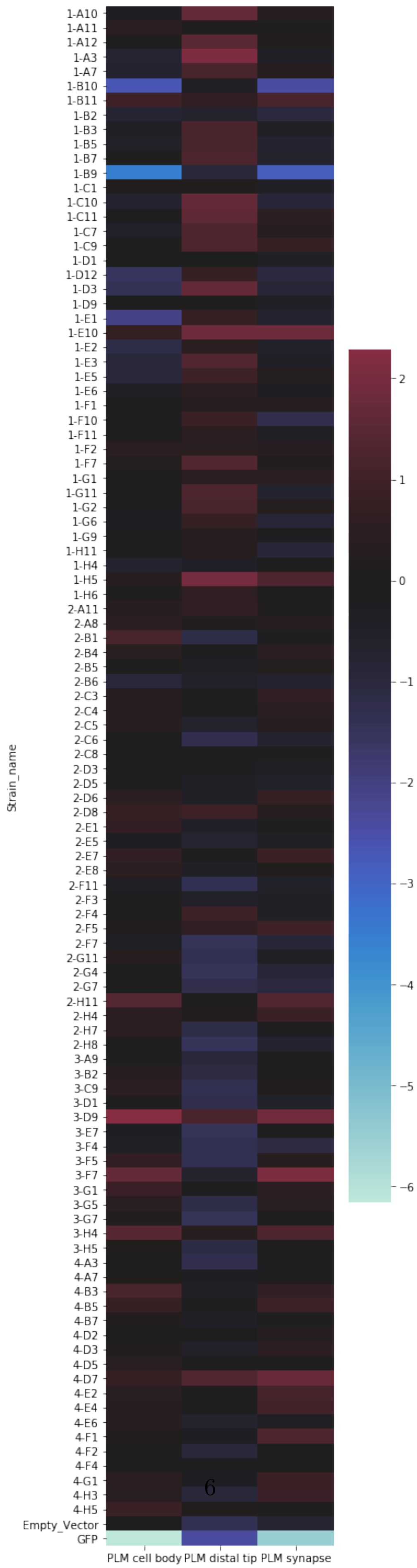


[6]:
```
df2 = df.replace(["low","low-med","med","med-high","high"], [int(0.0),int(1.
  ↪0),int(2.0),int(3.0),int(4.0)])
dfmean = df2.groupby('Strain_name', as_index=False).mean()

x = dfmean.loc[:, ['PLM cell body']]
dfmean['Intensity_z'] = (x)# - x.mean())/x.std()
dfmean['colors'] = ['red' if x < 0 else 'darkgreen' for x in␣
  ↪dfmean['Intensity_z']]
dfmean.sort_values('Intensity_z', inplace=True)
dfmean.reset_index(inplace=True)
#dfmean.set_index("Strain_name",drop=True,inplace=True)
```

```
plt.figure(figsize=(14,16), dpi= 80)
plt.scatter(dfmean.Intensity_z, dfmean.index, s=450, alpha=.6, color=dfmean.
  ↪colors)
for x, y, tex in zip(dfmean.Intensity_z, dfmean.index, dfmean.Intensity_z):
    t = plt.text(x, y, round(tex, 1), horizontalalignment='center',
                 verticalalignment='center', fontdict={'color':'white'})

plt.yticks(dfmean.index, dfmean.Strain_name)
#dfmean.plot.bar(y="PLM distal tip")
#plt.tight_layout()
plt.show()
```

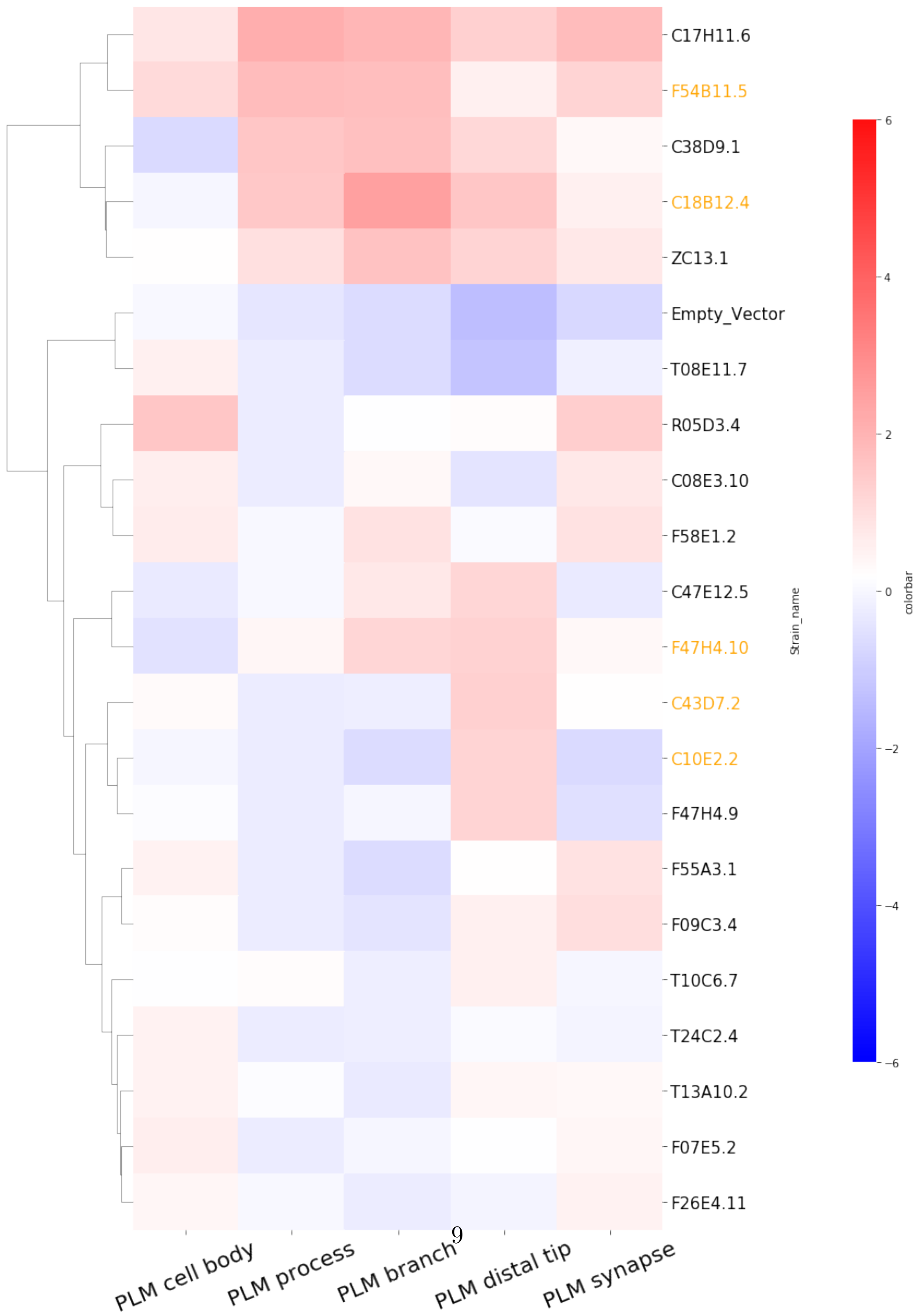


```
[21]: df2 = df.replace(["low","low-med","med","med-high","high"], [int(0.0),int(1.
       ↪0),int(2.0),int(3.0),int(4.0)])
      dfmean = df2.groupby('Strain_name', as_index=False).mean()

      x = dfmean.loc[:, ['PLM distal tip']]
      dfmean.set_index("Strain_name",drop=True,inplace=True)
      #normalized_df=(dfmean-dfmean.min())/(dfmean.max()-dfmean.min())
      normalized_df=(dfmean-dfmean.mean())/dfmean.std()
      #sns.heatmap(normalized_df[:-2], vmin=0.3, vmax=1)
      sns.heatmap(normalized_df[:-2].loc[:,["PLM cell body","PLM distal tip","PLM␣
       ↪synapse"]], yticklabels=True, center=0)

      '''dfmean['Intensity_z'] = (x)# - x.mean())/x.std()
      dfmean['colors'] = ['red' if x < 0 else 'darkgreen' for x in␣
       ↪dfmean['Intensity_z']]
      dfmean.sort_values('Intensity_z', inplace=True)
      dfmean.reset_index(inplace=True)
      #dfmean.set_index("Strain_name",drop=True,inplace=True)
      plt.figure(figsize=(14,16), dpi= 80)
      plt.scatter(dfmean.Intensity_z, dfmean.index, s=450, alpha=.6, color=dfmean.
       ↪colors)
      for x, y, tex in zip(dfmean.Intensity_z, dfmean.index, dfmean.Intensity_z):
          t = plt.text(x, y, round(tex, 1), horizontalalignment='center',
                       verticalalignment='center', fontdict={'color':'white'})

      plt.yticks(dfmean.index, dfmean.Strain_name)'''
      #dfmean.plot.bar(y="PLM distal tip")
      #plt.tight_layout()
      plt.subplots_adjust(top=5, left=0.3)
      #plt.savefig('D:\Data\RNAi_screen\RNAi-NII.png', dpi=400)
      plt.show()
```


1-A10
1-A11
1-A12
1-A3
1-A7
1-B10
1-B11
1-B2
1-B3
1-B5
1-B7
1-B9
1-C1
1-C10
1-C11
1-C7
1-C9
1-D1
1-D12
1-D3
1-D9
1-E1
1-E10
1-E2
1-E3
1-E5
1-E6
1-F1
1-F10
1-F11
1-F2
1-F7
1-G1
1-G11
1-G2
1-G6
1-G9
1-H11
1-H4
1-H5
1-H6
2-A11
2-A8
2-B1
2-B4
2-B5
2-B6
2-C3
2-C4
2-C5
2-C6
2-C8
2-D3
2-D5
2-D6
2-D8
2-E1
2-E5
2-E7
2-E8
2-F11
2-F3
2-F4
2-F5
2-F7
2-G11
2-G4
2-G7
2-H11
2-H4
2-H7
2-H8
3-A9
3-B2
3-C9
3-D1
3-D9
3-E7
3-F4
3-F5
3-F7
3-G1
3-G5
3-G7
3-H4
3-H5
4-A3
4-A7
4-B3
4-B5
4-B7
4-D2
4-D3
4-D5
4-D7
4-E2
4-E4
4-E6
4-F1
4-F2
4-F4
4-G1
4-H3
4-H5
Empty_Vector
GFP
Strain_name
PLM cell body
PLM distal tip
PLM synapse
2
1
0
−1
−2
−3
−4
−5
−6

[8]:
```
df2 = df.replace(["low","low-med","med","med-high","high"], [int(0.0),int(1.
↪0),int(2.0),int(3.0),int(4.0)]) #Replace intensity with numbers
dfmean = df2.groupby('Strain_name', as_index=False).mean() #Calculate average␣
↪for each column grouped by strain

#Strain to display
Everywhere = ("F54B11.5","C18B12.4","ZC13.1")
Synapse = ("T28B11.1","C49H3.5","K02A6.3","F09C3.4","F07E5.2","F26E4.
↪11","T08E11.7","F58E1.2","C08E3.10")
Distal = ("F47H4.9","C30F2.2","T10C6.7","D2085.4","C43D7.2","T24C2.4","F58H7.
↪7","T13A10.2","C10E2.2")
Gradient = ("ZK287.5","F47H4.10","F11A10.3","F26G5.9","C38D9.1")
All = Everywhere+Synapse+Distal+Gradient
Final = ("Empty_Vector","C47E12.5","R05D3.4","F55A3.1","C17H11.6","C43D7.
↪2","T13A10.2","F54B11.5","C38D9.1","T24C2.4","C18B12.4","C10E2.2","F47H4.
↪9","F47H4.10","ZC13.1","T08E11.7","T10C6.7","F07E5.2","F26E4.11","C08E3.
↪10","F58E1.2","F09C3.4")

#Strains to highlight
Control = ("Empty_Vector","GFP","C47E12.5")
E2 = ("C35B1.1","M7.1","Y71G12B.15","D1022.1","F58A4.10","Y94H6A.6","F29B9.
↪6","R09B3.4","Y54G2A.31","Y87G2A.9","Y110A2AR.2","Y54E5B.4","B0403.2","R01H2.
↪6","Y69H2.6","F40G9.3","C06E2.3","C06E2.7","C28G1.1","F49E12.4","F25H2.
↪8","Y110A2AM.3","F39B2.2","F56D2.4","F26H9.7")
mehta_genes = ("C26E6.5","F08G12.4","F22E12.4","F25B5.4","K11D2.1","T05H10.
↪5","Y47D7A.1","Y6B3A.1","ZK287.5","B0547.1","D2045.6","F36A2.13","F46A9.
↪5","F46A9.50","Y55F3AM.15","ZK520.4","C17H11.6","F45H11.2","C18B12.
↪4","K12C11.2","Y59A8A.1","T14G10.6","F35D6.1","C02F5.7","C30F2.2","C52D10.
↪7","K08E7.7","F53C11.8","D2085.4","T09B4.10","C10E2.2","K04G11.4")
No_orthologues = ("T28B11.1","K02A6.3","F09C3.4","F07E5.2","T08E11.7","C08E3.
↪10","T10C6.7","F47H4.9","F58H7.7")
Orthologues_unknown_fn = ("F54B11.5","C18B12.4","C10E2.2","F47H4.10","C43D7.2")
Transport = ("T24C2.4","T13A10.2","F58E1.2","ZC13.1")
Repressor = ("C49H3.5","C30F2.2","F26G5.9","F11A10.3","C38D9.1")

#Columns to display
xticks_ = ["PLM cell body","PLM process","PLM branch","PLM distal tip","PLM␣
↪synapse"]#,"ALM cell body","ALM process","ALM branch","ALM distal tip","ALM␣
↪synapse"]
yticks_ = ['F47H4.9','C30F2.2','T10C6.7','D2085.4','C43D7.2','T24C2.4','F58H7.
↪7','T13A10.2','C10E2.2']

x = dfmean.loc[:, ['PLM distal tip']]
dfmean.set_index("Strain_name",drop=True,inplace=True)
```

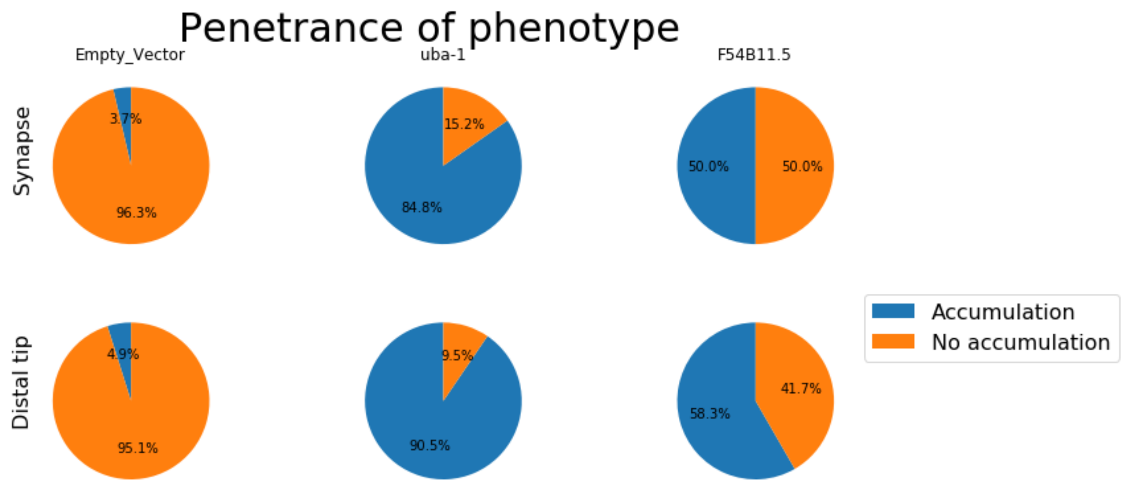
```
#Normalization criteria
#normalized_df=(dfmean-dfmean.min())/(dfmean.max()-dfmean.min())
normalized_df=(dfmean-dfmean.mean())/dfmean.std()

#Plot heatmap
#sns.heatmap(normalized_df[:-2], vmin=0.3, vmax=1)
#sns.heatmap(normalized_df[:-2], center=0).loc[All,xticks_]

#print(normalized_df[:-2].to_string())
g = sns.clustermap(normalized_df[:-2].loc[Final,xticks_], center=0.2,␣
 ↪yticklabels=True, vmin=-6, vmax=+6,cbar_kws={'label': 'colorbar'},
                   figsize=(10, 20),col_cluster=False, cmap='bwr',␣
 ↪method="complete", metric="euclidean")

plt.setp(g.ax_heatmap.xaxis.get_majorticklabels(), rotation=23, size=20)
plt.setp(g.ax_heatmap.yaxis.get_majorticklabels(), rotation=0, size=15)
g.cax.set_position((1.1,0.126,.03,0.6))

for tick_label in g.ax_heatmap.axes.get_yticklabels():
    if tick_label.get_text() in No_orthologues:
        tick_label.set_color("pink")
    if tick_label.get_text() in Transport:
        tick_label.set_color("green")
    if tick_label.get_text() in Repressor:
        tick_label.set_color("blue")
    if tick_label.get_text() in Orthologues_unknown_fn:
        tick_label.set_color("orange")
    else:
        tick_label.set_color("black")

plt.savefig(path + '/' + 'heatmap_selected_PLM_new-2.png', bbox_inches="tight",␣
 ↪transparent=True)
plt.show()
```


C17H11.6
F54B11.5
C38D9.1
C18B12.4
ZC13.1
Empty_Vector
T08E11.7
R05D3.4
C08E3.10
F58E1.2
C47E12.5
F47H4.10
C43D7.2
C10E2.2
F47H4.9
F55A3.1
F09C3.4
T10C6.7
T24C2.4
T13A10.2
F07E5.2
F26E4.11
PLM cell body
PLM process
PLM branch
PLM distal tip
PLM synapse
Strain_name
colorbar
6
4
2
0
−2
−4
−6

```
[1]: Everywhere = ("F54B11.5","C18B12.4","ZC13.1")
     Synapse = ("T28B11.1","C49H3.5","K02A6.3","F09C3.4","F07E5.2","F26E4.
      ↪11","T08E11.7","F58E1.2","C08E3.10")
     Distal = ("F47H4.9","C30F2.2","T10C6.7","D2085.4","C43D7.2","T24C2.4","F58H7.
      ↪7","T13A10.2","C10E2.2")
     Gradient = ("ZK287.5","F47H4.10","F11A10.3","F26G5.9","C38D9.1")
     All = Everywhere+Synapse+Distal+Gradient
     print(All)
```

```
('F54B11.5', 'C18B12.4', 'ZC13.1', 'T28B11.1', 'C49H3.5', 'K02A6.3', 'F09C3.4',
'F07E5.2', 'F26E4.11', 'T08E11.7', 'F58E1.2', 'C08E3.10', 'F47H4.9', 'C30F2.2',
'T10C6.7', 'D2085.4', 'C43D7.2', 'T24C2.4', 'F58H7.7', 'T13A10.2', 'C10E2.2',
'ZK287.5', 'F47H4.10', 'F11A10.3', 'F26G5.9', 'C38D9.1')
```

```
[115]: df2 = df.replace(["low","low-med","med","med-high","high"], [int(0.0),int(1.
        ↪0),int(2.0),int(3.0),int(4.0)]) #Replace intensity with numbers
       fig, ax = plt.subplots(2,3, figsize=(12, 6), subplot_kw=dict(aspect="equal"))

       for i in enumerate(['Empty_Vector','uba-1','F54B11.5']):
           total_n=[]
           Pen_n=[]
           total_n=(df2.loc[(df2['Strain_name'] == i[1])]).count()[0]#[df2['PLM distal␣
        ↪tip'] > 3].count()[0]
           Pen_tip=(df2.loc[(df2['Strain_name'] == i[1])&(df2['PLM distal tip'] > 1)]).
        ↪count()[0]#[df2['PLM distal tip'] > 3].count()[0]
           Pen_syn=(df2.loc[(df2['Strain_name'] == i[1])&(df2['PLM synapse'] > 2)]).
        ↪count()[0]#[df2['PLM distal tip'] > 3].count()[0]
       #     print(t)
           patches, texts, junk = ax[0,i[0]].pie([Pen_syn,total_n-Pen_syn],␣
        ↪autopct='%1.1f%%', startangle = 90)#, colors = mycolors)
           patches, texts, junk = ax[1,i[0]].pie([Pen_tip,total_n-Pen_tip],␣
        ↪autopct='%1.1f%%', startangle = 90)#, colors = mycolors)
           ax[0,i[0]].set_title(i[1])
       ax[0,0].text(-1.5,-.3,'Synapse', rotation=90,fontsize=16)
       ax[1,0].text(-1.5,-.3,'Distal tip', rotation=90,fontsize=16)
           #     ax[1,i[0]].set_title("Distal tip")
       fig.patch.set_facecolor('white')
       plt.legend(labels = ['Accumulation','No accumulation'], bbox_to_anchor=(1, 1.
        ↪1),fontsize=16)
       plt.suptitle("Penetrance of phenotype",fontsize=30)
       plt.savefig(path + '/' + 'Penetrance_accumulation_PLM.png',␣
        ↪bbox_inches="tight", transparent=True)

       plt.show()
```



```
[116]: path
```

```
[116]: 'D:\\Data\\RNAi_screen'
```

```
[ ]:
```