ATAC_and_CRISPR_score

November 27, 2017

1 This workbook will be used to analyze any correlations between ATAC seq peaks and CRISPR screen scores.

Let's start by reading the data into a variable called df.

```
In [1]: import pandas as pd
        import numpy as np
        df = pd.read_excel \
        (r'/Users/eatonaw/Desktop/Atom_Files/ATAC/2017-10-24-CIV_by_ATAC2.xlsx')
        print(df.head())
        print(df.columns)
```

```
  Chromosome     Start      Stop  TMD8_CIV2_D21_CSS  TMD8_CIV2_IBR_CSS  \
0      chr19  40791075  40791094          -0.731734              -1.15
1      chr19  40791166  40791185          -0.602970              -1.21
2      chr19  40791167  40791186          -0.005454              -2.05
3      chr19  40791302  40791321           0.399748              -0.82
4      chr19  40791220  40791239           1.168600              -3.56

  ATAC_tmd8_chr     start        stop         Peak name Score.Peak.color  \
0         chr19  40790622  40791786.0  TMD8_peak_545090              3249
1         chr19  40790622  40791786.0  TMD8_peak_545090              3249
2         chr19  40790622  40791786.0  TMD8_peak_545090              3249
3         chr19  40790622  40791786.0  TMD8_peak_545090              3249
4         chr19  40790622  40791786.0  TMD8_peak_545090              3249

  strand signalValue negLog10pval qval(-log10FDR) peak  \
0      .     21.2455      324.952         320.267  843
1      .     21.2455      324.952         320.267  843
2      .     21.2455      324.952         320.267  843
3      .     21.2455      324.952         320.267  843
4      .     21.2455      324.952         320.267  843

  uniqu Match for vlookup GeneSybmbol TMD8_Brun_CSS Essential Fig1 gene
0   chr194079107540791094        AKT2       -0.4339       NaN      AKT2
1   chr194079116640791185        AKT2       -0.4339       NaN      AKT2
2   chr194079116740791186        AKT2       -0.4339       NaN      AKT2
```

```
3   chr194079130240791321        AKT2       -0.4339       NaN      AKT2
4   chr194079122040791239        AKT2       -0.4339       NaN      AKT2
Index(['Chromosome', 'Start', 'Stop', 'TMD8_CIV2_D21_CSS', 'TMD8_CIV2_IBR_CSS',
       'ATAC_tmd8_chr', 'start', 'stop', 'Peak name', 'Score.Peak.color',
       'strand', 'signalValue', 'negLog10pval', 'qval(-log10FDR)', 'peak',
       'uniqu Match for vlookup', 'GeneSybmbol', 'TMD8_Brun_CSS', 'Essential',
       'Fig1 gene'],
      dtype='object')
```

First, let's format our data from df into a new variable called df2. This will only include a subset of columns from df that we are interested in. In addition, we will change the data types of some of the columns and format it for an easier way to analyze the data.

```
In [2]: df.rename(columns = {'Score.Peak.color' : 'Color_Score', 'signalValue':\
         'Signal_Value', 'GeneSybmbol' : 'Gene_Symbol', 'Peak Name' : 'Peak_Name',\
         'uniqu Match for vlookup' : 'Unique', 'Fig1 gene': 'Fig1'}, inplace = True)

        df2 = df[['Chromosome', 'TMD8_CIV2_D21_CSS', 'TMD8_CIV2_IBR_CSS', 'Color_Score',\
         'Signal_Value', 'Unique', 'Gene_Symbol', 'TMD8_Brun_CSS', 'Essential', 'Fig1']]

        df2.loc[:,['Color_Score','Signal_Value', 'TMD8_Brun_CSS']] = df2.loc[:,['Color_Score'\
         ,'Signal_Value', 'TMD8_Brun_CSS']].apply(pd.to_numeric, errors = 'coerce')

        print(df2.dtypes)
        df2.head()
```

```
Chromosome            object
TMD8_CIV2_D21_CSS    float64
TMD8_CIV2_IBR_CSS    float64
Color_Score          float64
Signal_Value         float64
Unique                object
Gene_Symbol           object
TMD8_Brun_CSS        float64
Essential             object
Fig1                  object
dtype: object


/usr/local/lib/python3.6/site-packages/pandas/core/indexing.py:517: 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/indexing.html
  self.obj[item] = s
```

Out[2]:

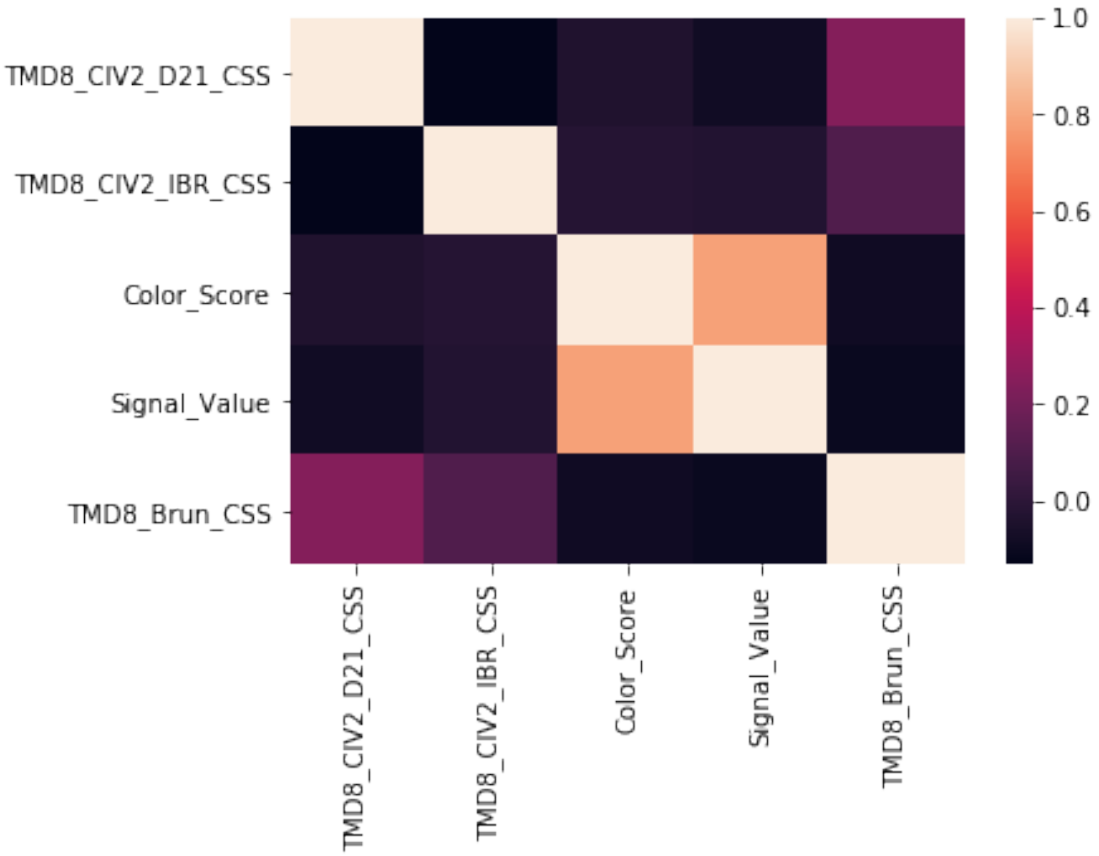
	Chromosome	TMD8_CIV2_D21_CSS	TMD8_CIV2_IBR_CSS	Color_Score	Signal_Value	\
0	chr19	-0.731734	-1.15	3249.0	21.24547	
1	chr19	-0.602970	-1.21	3249.0	21.24547	
2	chr19	-0.005454	-2.05	3249.0	21.24547	
3	chr19	0.399748	-0.82	3249.0	21.24547	
4	chr19	1.168600	-3.56	3249.0	21.24547	

	Unique	Gene_Symbol	TMD8_Brun_CSS	Essential	Fig1
0	chr194079107540791094	AKT2	-0.4339	NaN	AKT2
1	chr194079116640791185	AKT2	-0.4339	NaN	AKT2
2	chr194079116740791186	AKT2	-0.4339	NaN	AKT2
3	chr194079130240791321	AKT2	-0.4339	NaN	AKT2
4	chr194079122040791239	AKT2	-0.4339	NaN	AKT2

Let's start off with some visualizing to see if we can see any interesting correlations or anything else in the data set.

```
In [3]: import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
        sns.heatmap(df2.corr())
```

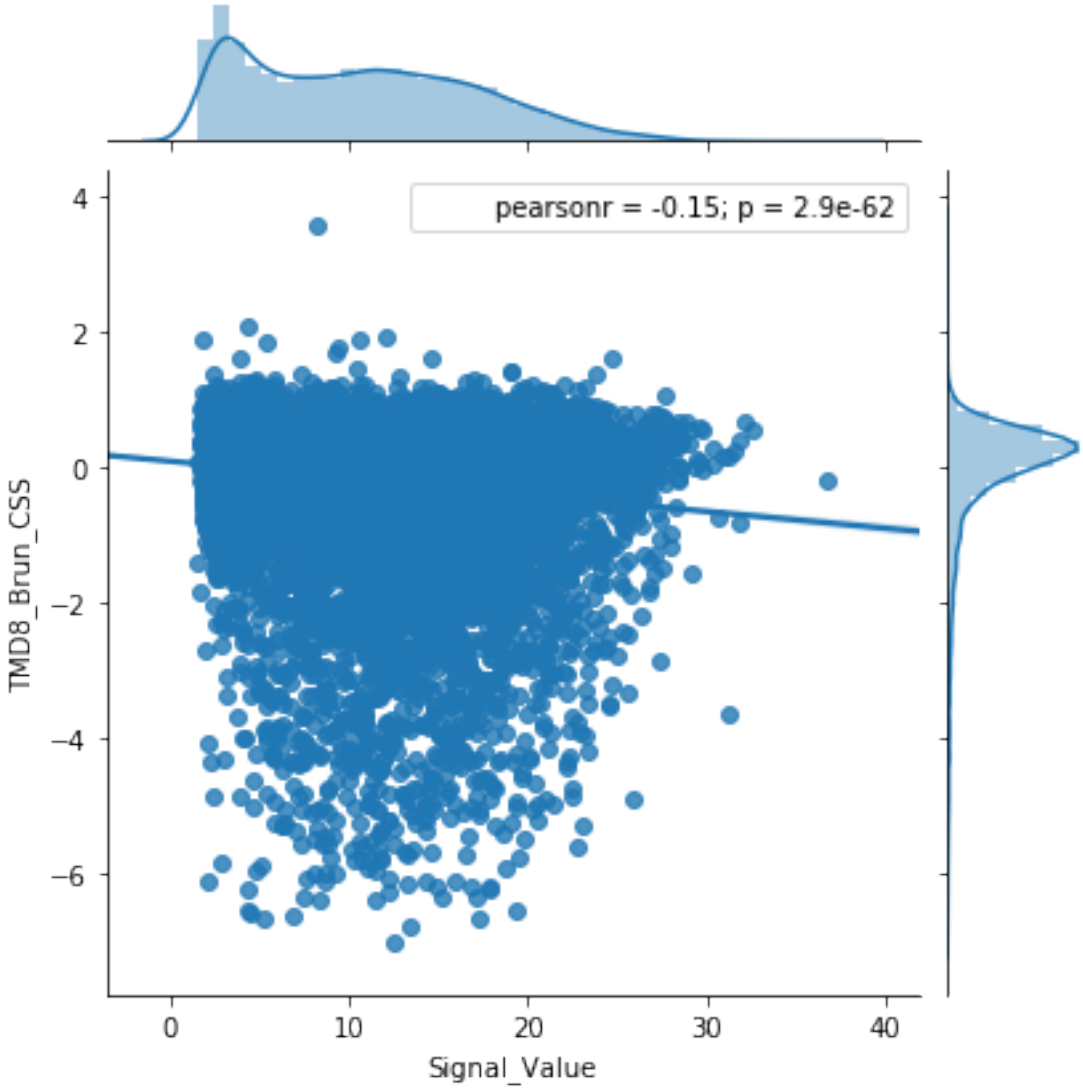
```
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x11038a0f0>
```

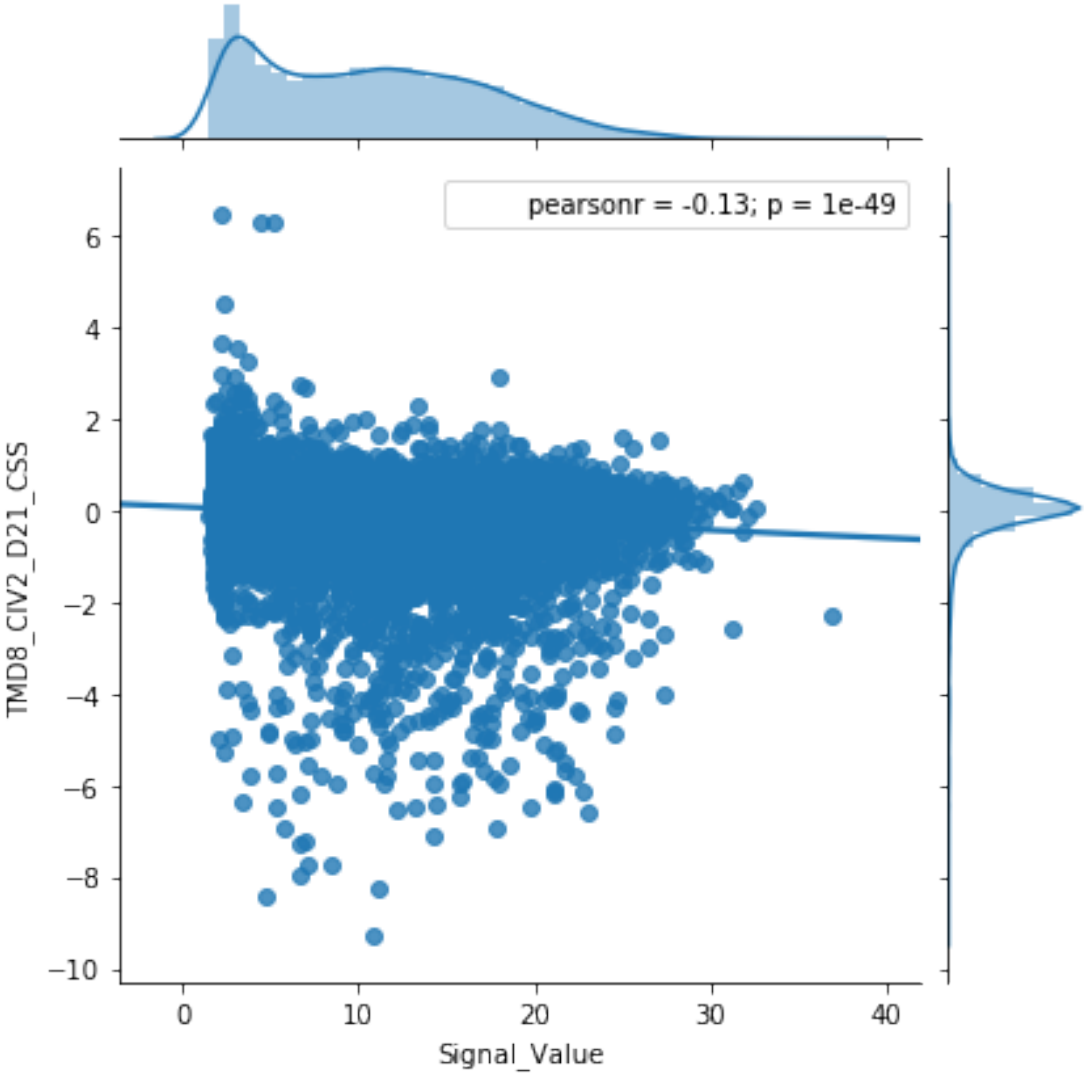


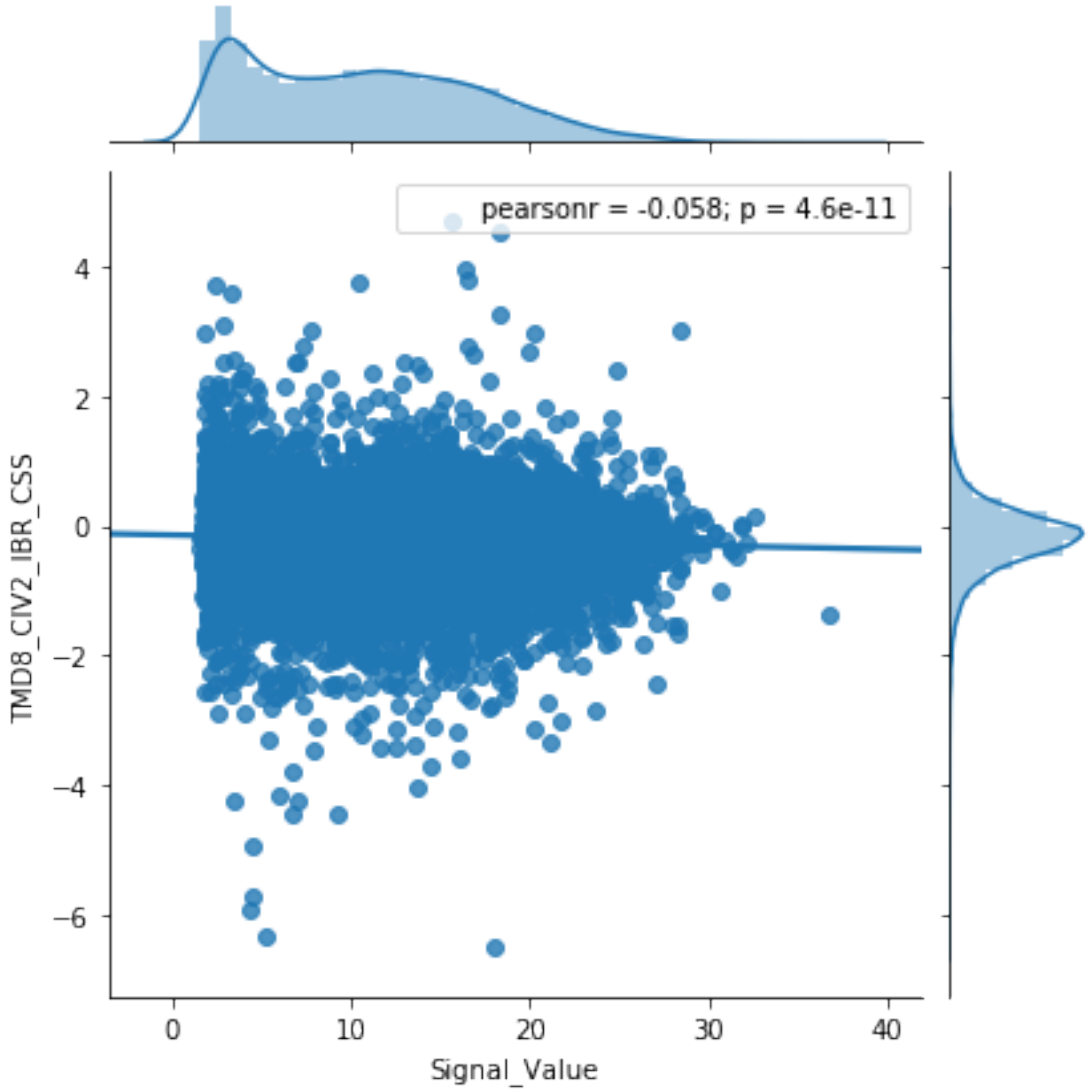
Here, we can see that color score and signal value are really correlated, so they are probably interchangable ways to quantify the ATAC seq signal data.

```
In [4]: sns.jointplot(x = 'Signal_Value', y = 'TMD8_Brun_CSS', data = df2.dropna\
         (subset = ['Signal_Value']).groupby('Gene_Symbol').mean(), kind = 'reg')

        sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_D21_CSS', data = df2.dropna\
         (subset = ['Signal_Value']).groupby('Gene_Symbol').mean(), kind = 'reg')

        sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_IBR_CSS', data = df2.dropna\
         (subset = ['Signal_Value']).groupby('Gene_Symbol').mean(), kind = 'reg')
```

```
Out[4]: <seaborn.axisgrid.JointGrid at 0x111a78080>
```



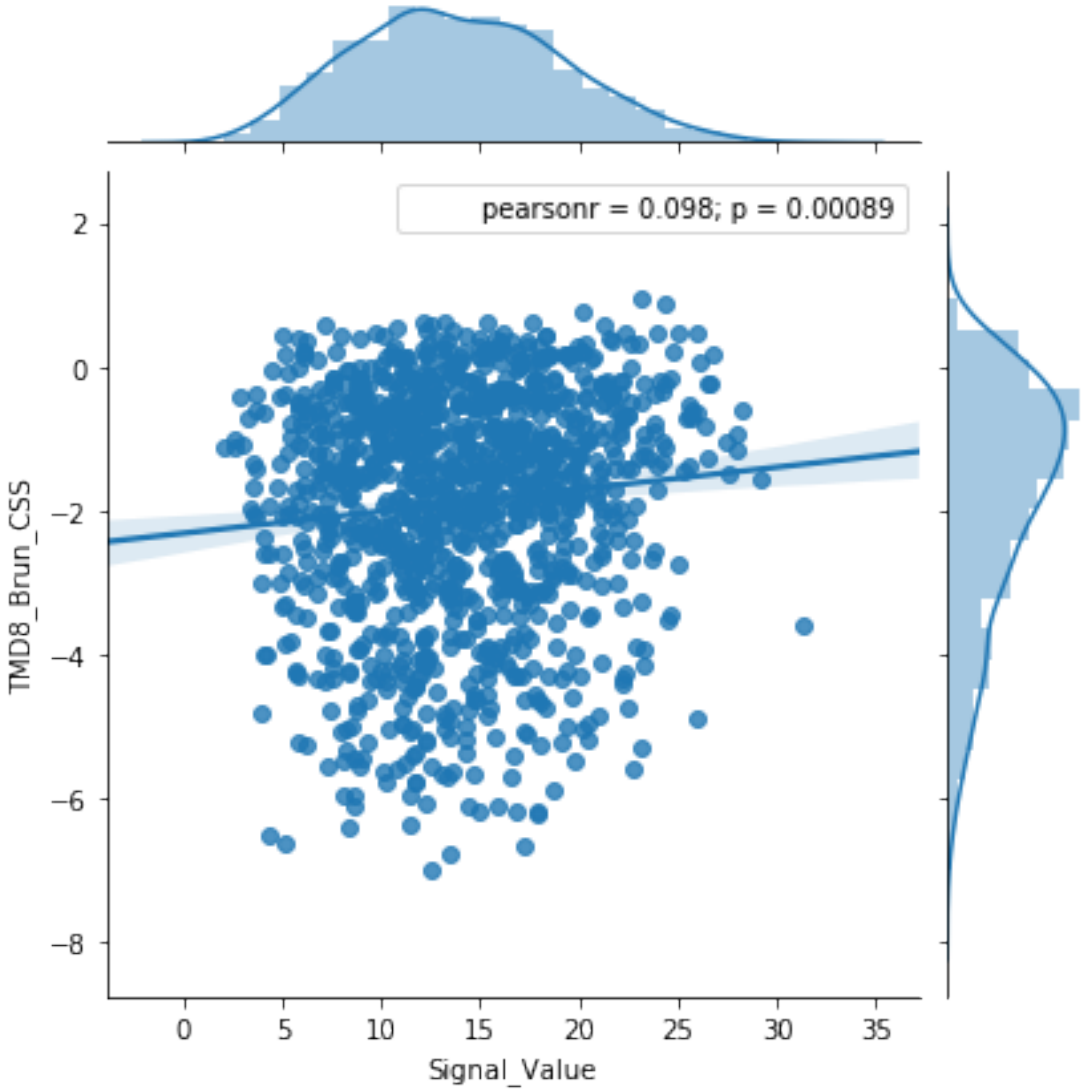

pearsonr = -0.13; p = 1e-49
TMD8_CIV2_D21_CSS
Signal_Value
6
4
2
0
−2
−4
−6
−8
−10
0
10
20
30
40

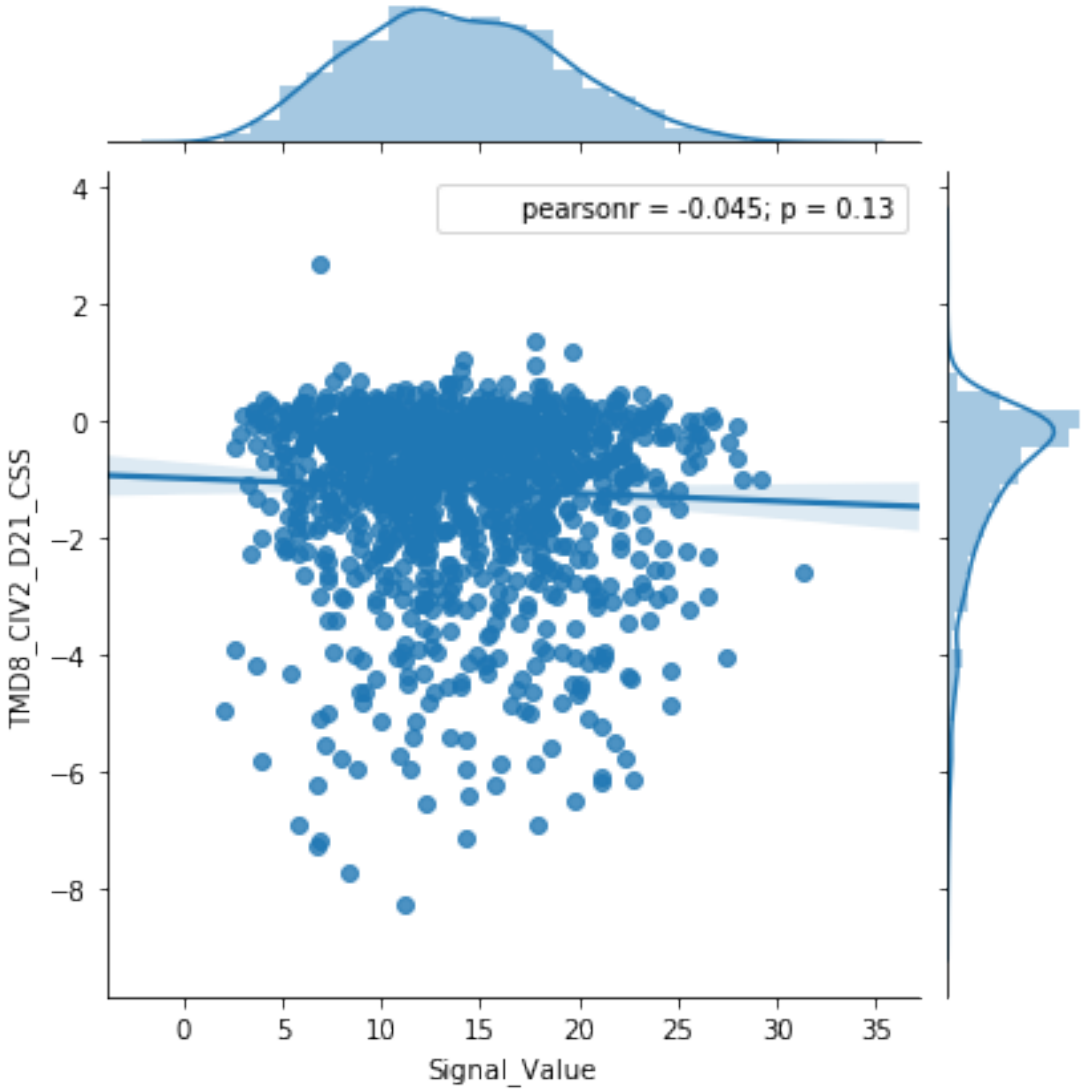


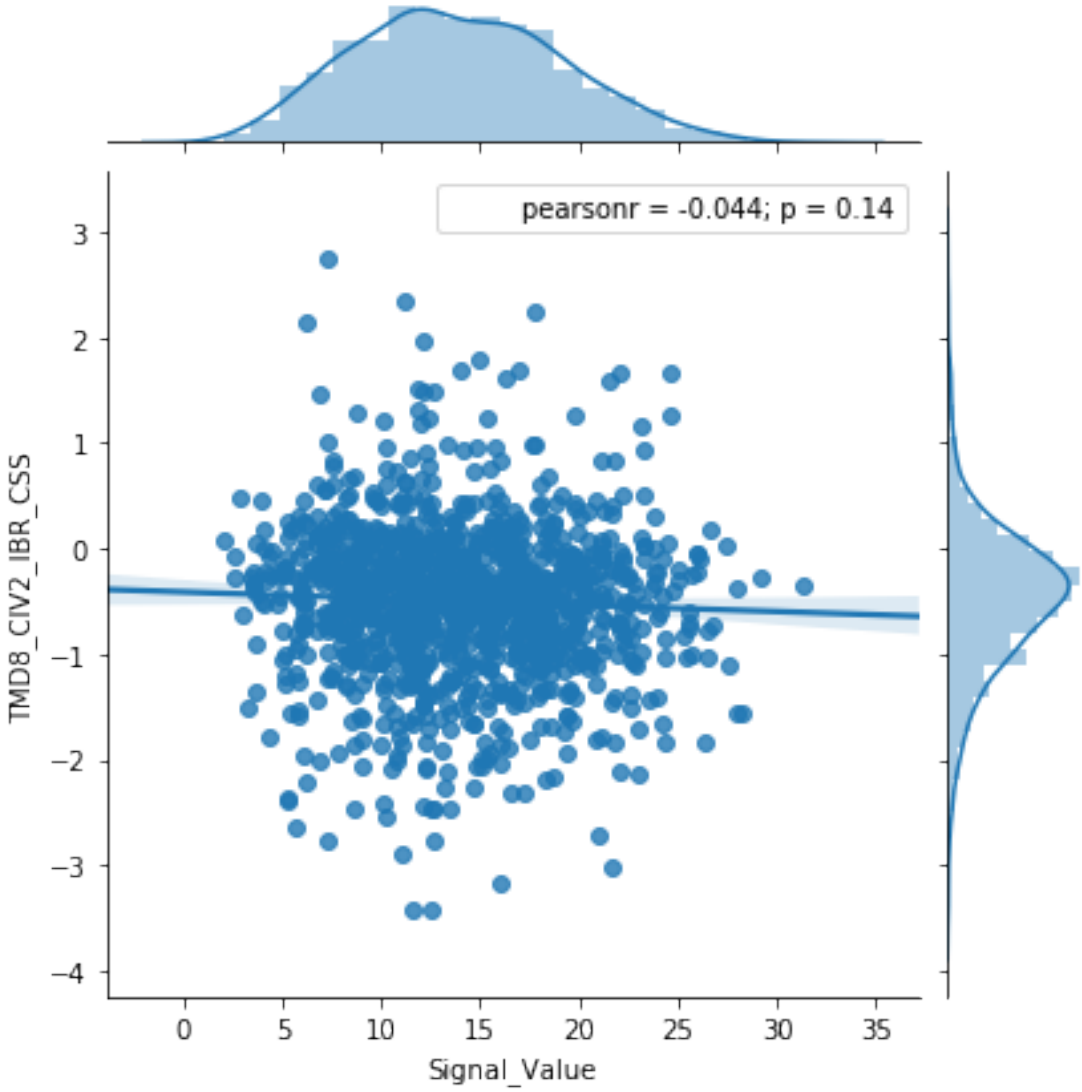
Unfortunately, there is no easily discernible trend between our CRISPR screening scores and ATAC data. Let's try subsetting the data. First, we will look at "Essential" genes.

```
In [5]: sns.jointplot(x = 'Signal_Value', y = 'TMD8_Brun_CSS', data = df2.dropna\
         (subset = ['Signal_Value', 'Essential']).groupby('Gene_Symbol').mean(), kind = 'reg')

        sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_D21_CSS', data = df2.dropna\
         (subset = ['Signal_Value', 'Essential']).groupby('Gene_Symbol').mean(), kind = 'reg')

        sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_IBR_CSS', data = df2.dropna\
         (subset = ['Signal_Value', 'Essential']).groupby('Gene_Symbol').mean(), kind = 'reg')
```

```
Out[5]: <seaborn.axisgrid.JointGrid at 0x11129e320>
```


pearsonr = 0.098; p = 0.00089
TMD8_Brun_CSS
Signal_Value
2
0
−2
−4
−6
−8
0
5
10
15
20
25
30
35

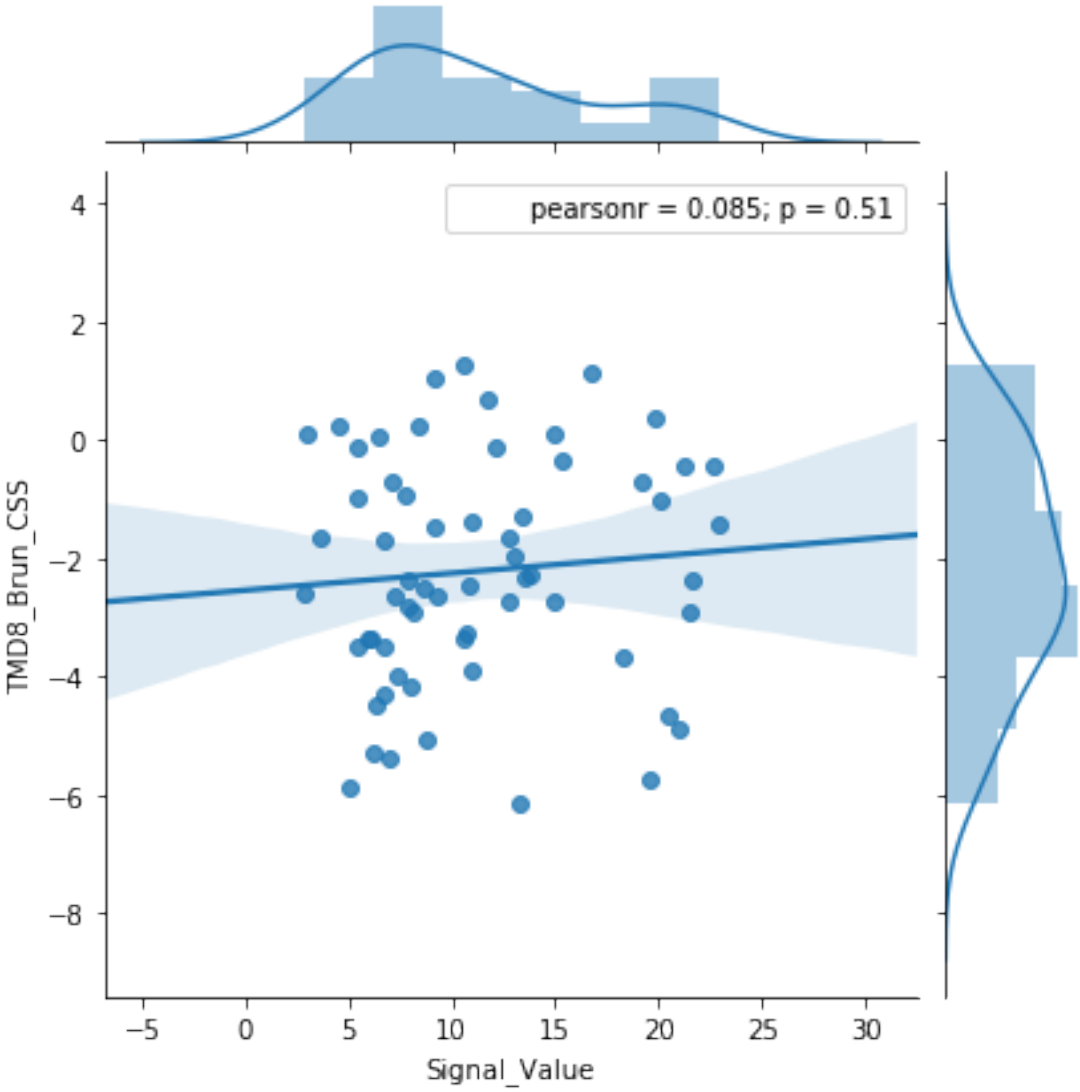

pearsonr = -0.045; p = 0.13
TMD8_CIV2_D21_CSS
Signal_Value
4
2
0
−2
−4
−6
−8
0
5
10
15
20
25
30
35

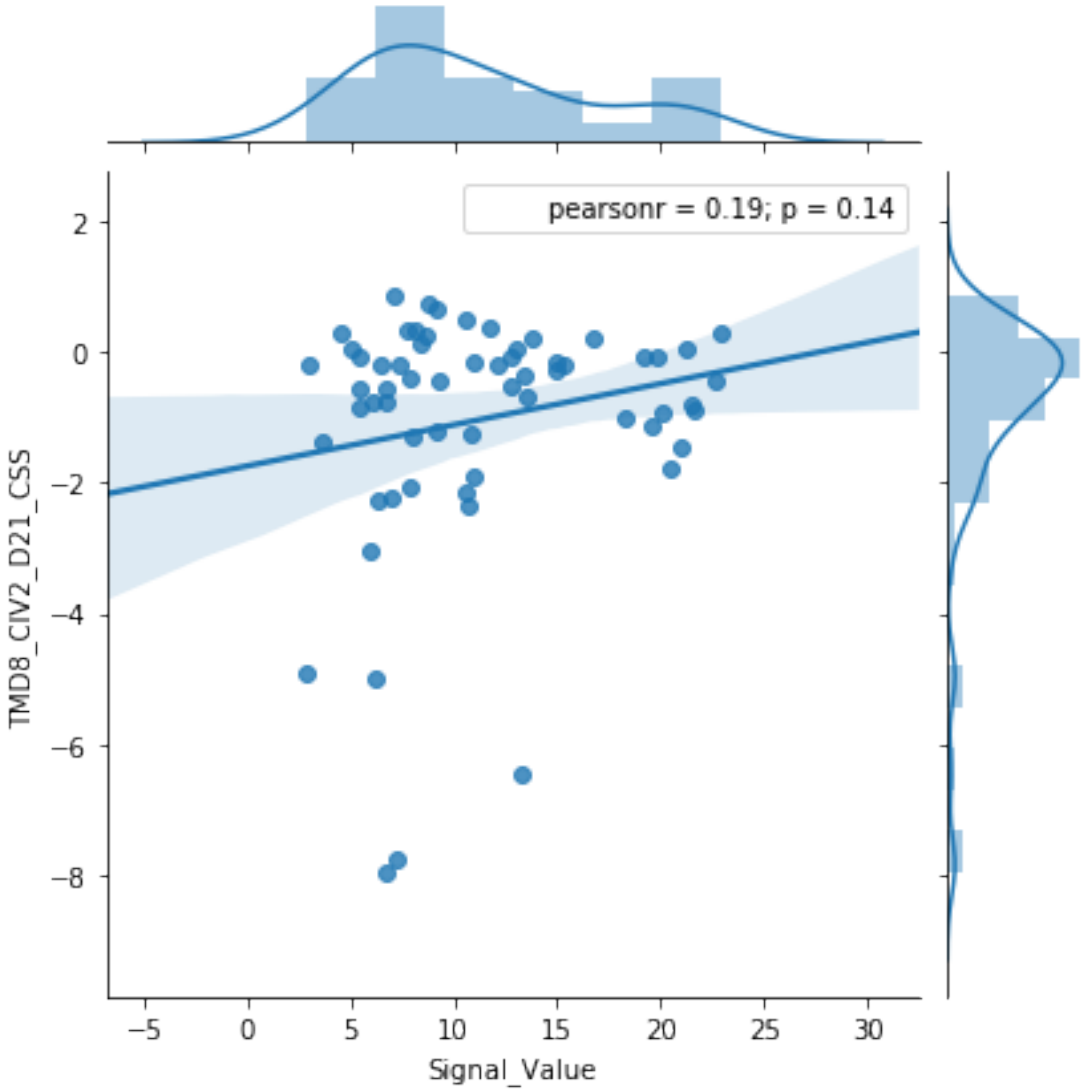


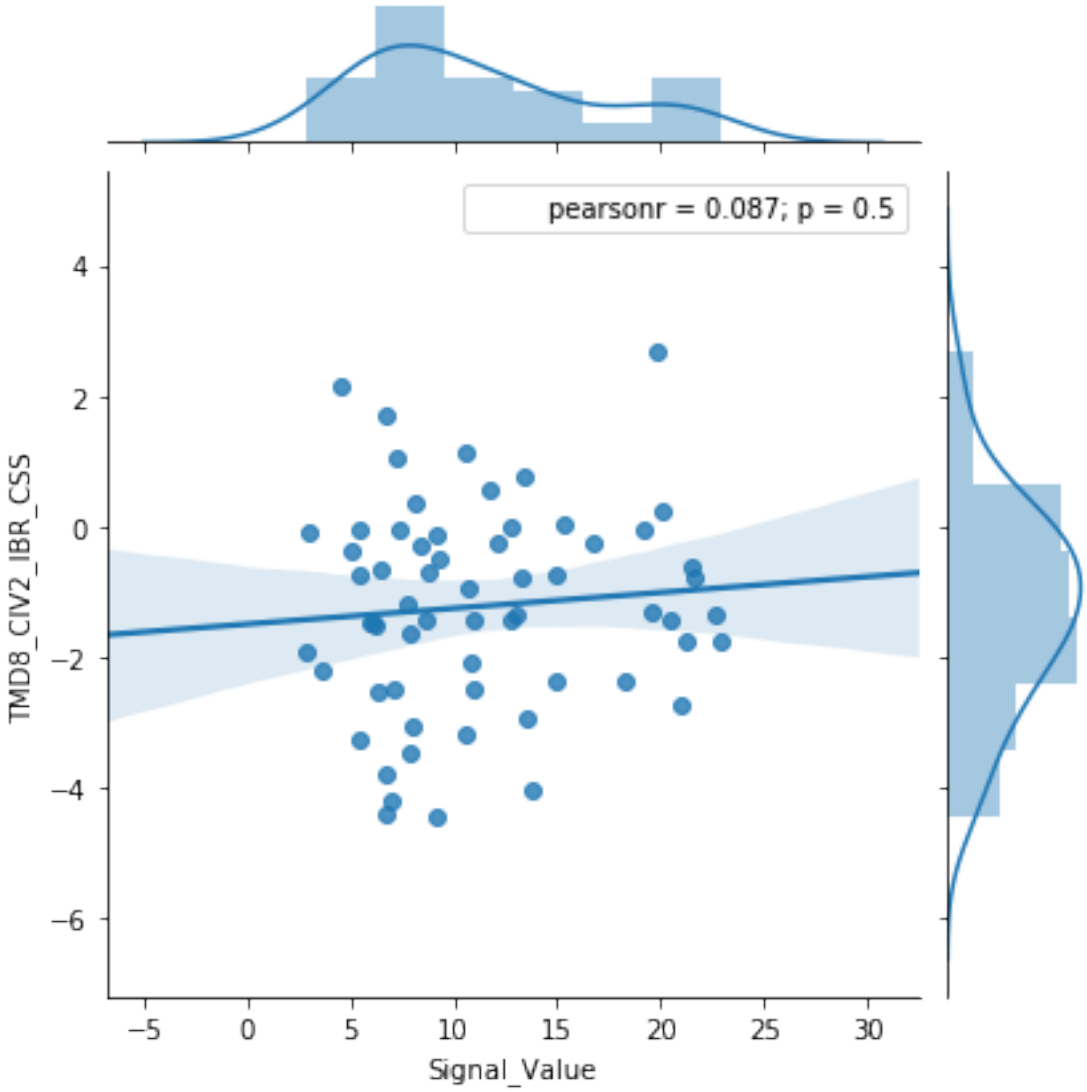
Unfortunately, no interesting conclusions arise from this. Let's now try this with "Figure 1" genes.

```
In [6]: sns.jointplot(x = 'Signal_Value', y = 'TMD8_Brun_CSS', data = df2.dropna\
         (subset = ['Signal_Value', 'Fig1']).groupby('Gene_Symbol').mean(), kind = 'reg')

        sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_D21_CSS', data = df2.dropna\
         (subset = ['Signal_Value', 'Fig1']).groupby('Gene_Symbol').mean(), kind = 'reg')

        sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_IBR_CSS', data = df2.dropna\
         (subset = ['Signal_Value', 'Fig1']).groupby('Gene_Symbol').mean(), kind = 'reg')
```

```
Out[6]: <seaborn.axisgrid.JointGrid at 0x111fa9630>
```


pearsonr = 0.085; p = 0.51
TMD8_Brun_CSS
Signal_Value
4
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25
30

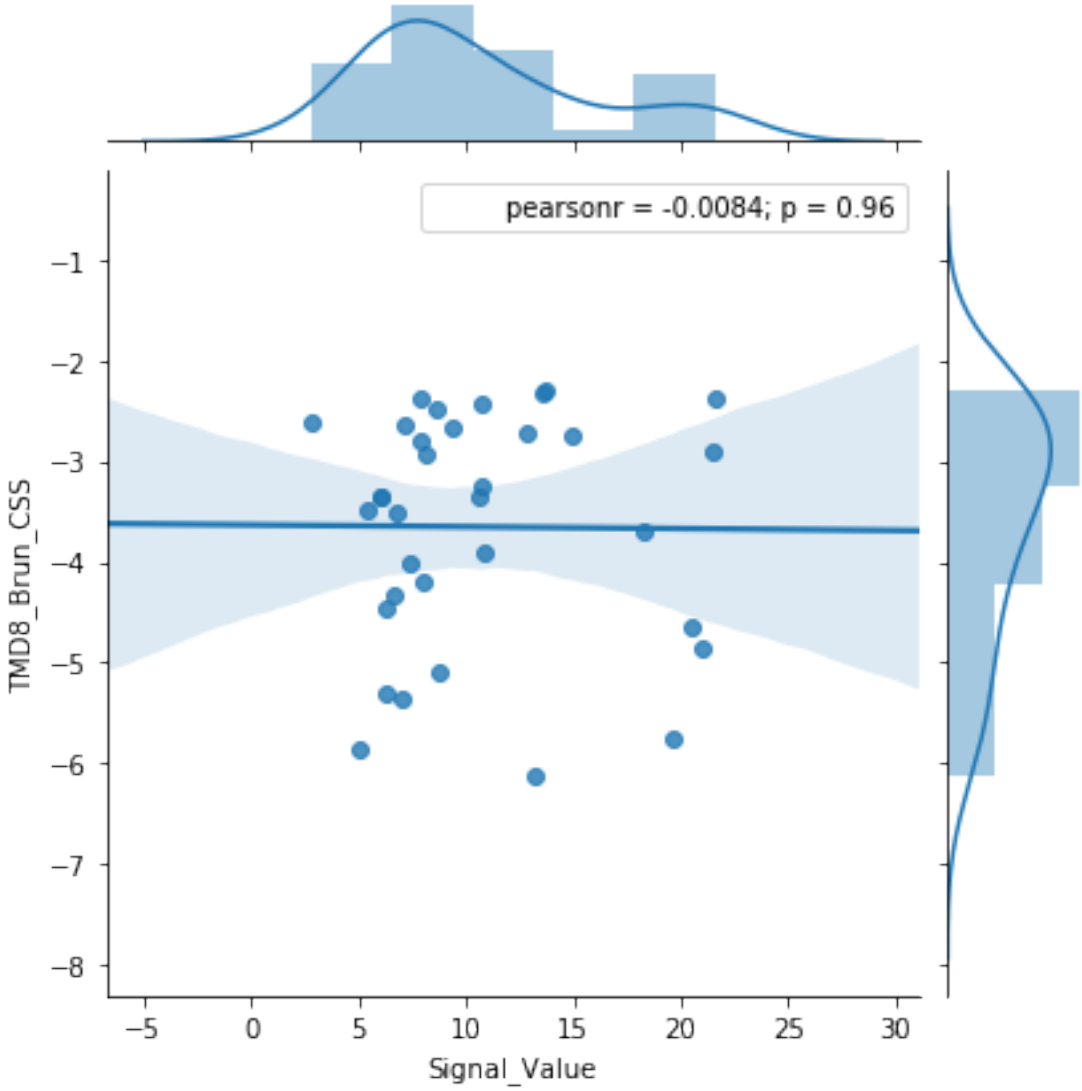

pearsonr = 0.19; p = 0.14
TMD8_CIV2_D21_CSS
Signal_Value
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25
30

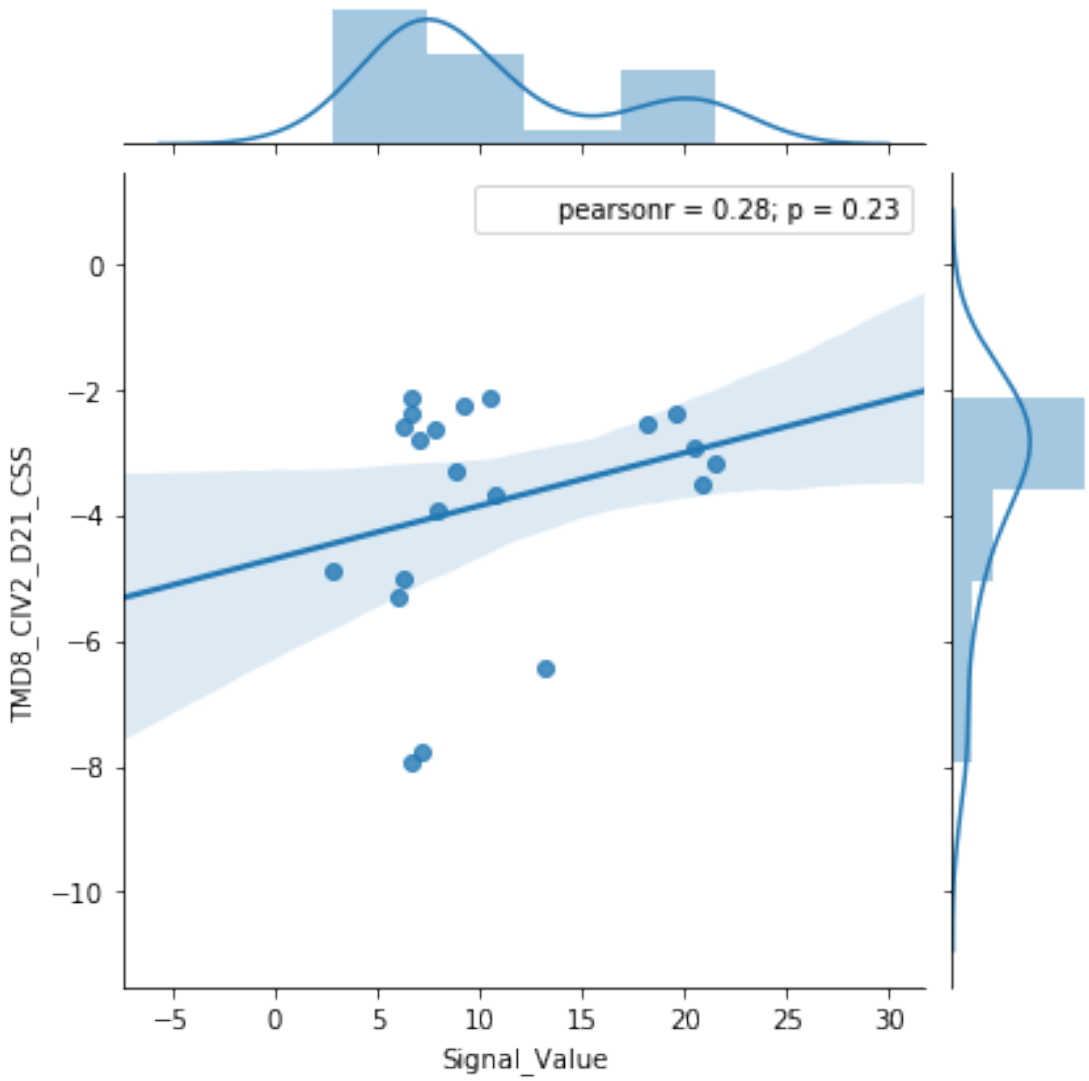


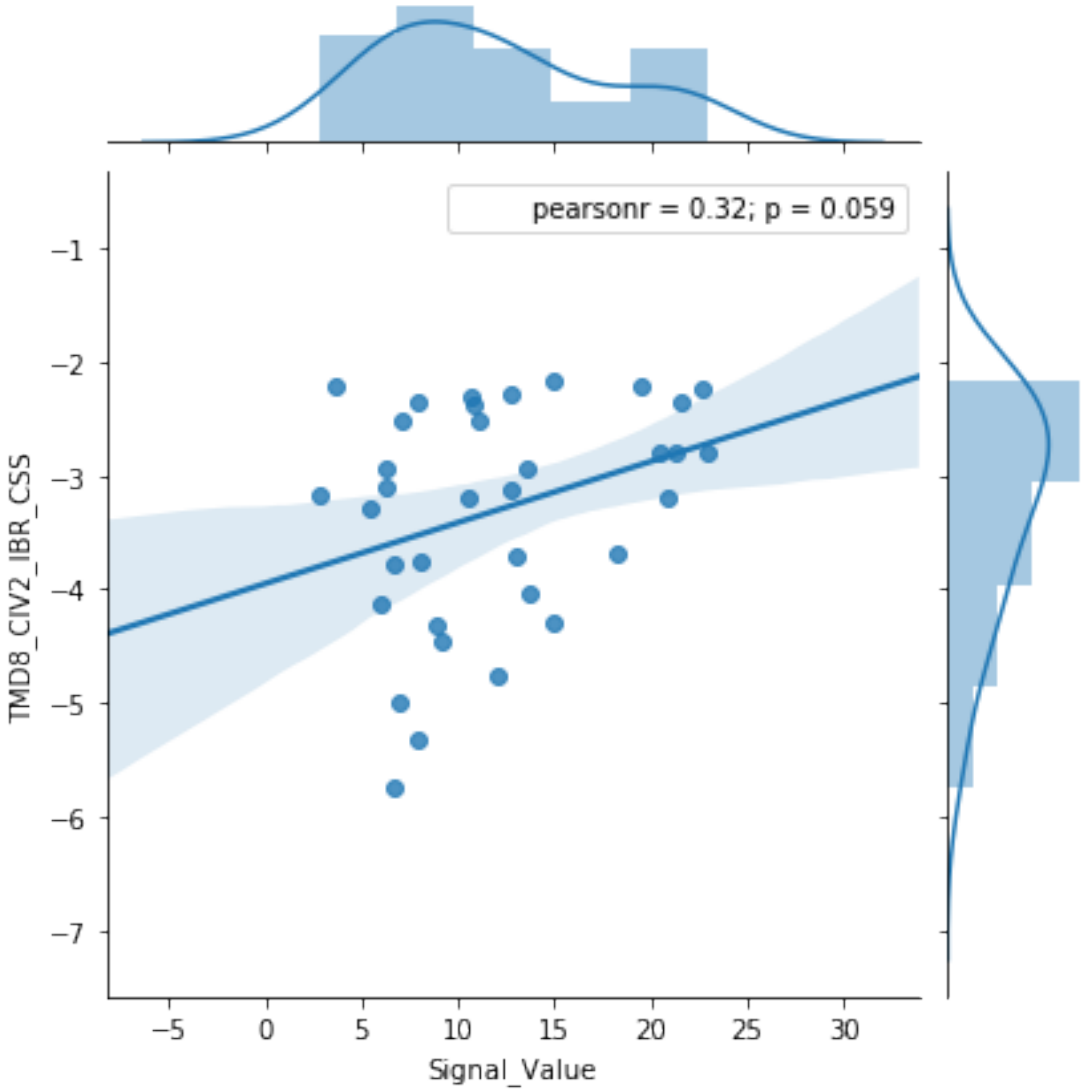
Let's now look at the figure 1 genes where the CRISPR score was < -2.

```
In [41]: sns.jointplot(x = 'Signal_Value', y = 'TMD8_Brun_CSS', data = df2\
         [df2.TMD8_Brun_CSS < -2].dropna(subset = ['Signal_Value', 'Fig1']).groupby\
         ('Gene_Symbol').mean(), kind = 'reg')

         sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_D21_CSS', data = df2\
         [df2.TMD8_CIV2_D21_CSS < -2].dropna(subset = ['Signal_Value', 'Fig1']).groupby\
         ('Gene_Symbol').mean(), kind = 'reg')

         sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_IBR_CSS', data = df2\
         [df2.TMD8_CIV2_IBR_CSS < -2].dropna(subset = ['Signal_Value', 'Fig1']).groupby\
         ('Gene_Symbol').mean(), kind = 'reg')
```

```
Out[41]: <seaborn.axisgrid.JointGrid at 0x11c290e80>
```




pearsonr = 0.28; p = 0.23
TMD8_CIV2_D21_CSS
Signal_Value
0
−2
−4
−6
−8
−10
−5
0
5
10
15
20
25
30



We can see here that there is no correlation between ATAC signal and CRISPR cutting score. However, we do see a slight correlation between ATAC signal CRISPRi score. There is an direct relationship here - the lower the ATAC signal, the lower the CRISPR score. This suggests that it may be harder to silence highly transcribed genes using the CRISPRi system.