ATAC_and_CRISPR_score

December 1, 2017

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

ATAC-seq (short for Assay for Transposase-Accessible Chromatin using sequencing) is an experimental method that quantifies the chromatim accessibility of a gene. In simplified terms, the higher the ATAC-seq signal for a certain gene, the higher that gene is transcribed (aka more expressed/turned on).

The CRISPR/CRISPRi screen scores come from two similar technologies that work in different ways. CRISPR (the Brunello screen) works by cutting a gene in its coding region, which disables it from functioning properly. CRISPRi works by binding to the transcriptional start site and inhibiting transcription/expression of the gene. For both methodologies, a negative CRISPR screen score (CSS) for a gene indicates that cells that lost the gene were less fit than cells that did not lose that gene.

When examining our CRISPR and CRISPRi data, we noticed some genes did not perform as well in CRISPRi in DMSO as we had predicted based on prior knowledge of essential genes in DLBCL. However, some of these genes performed well in Ibrutinib (IBR). We therefore hypothesized that this may be because it is hard to inhibit transcription of a highly expressed gene (high ATAC-seq signal) through CRISPRi. Ibrutinib possibly primes to cell to grow more slowly or lowers basal transcription of some genes, allowing for CRISPRi to work better. This workbook is an attempt to get to the bottom of this.

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

```
In [36]: 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 [67]: 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.Chromosome = df2.Chromosome.str[3:]
         df2.Chromosome.replace('X', 23, inplace=True)
         df2.Chromosome.replace('Y', 24, inplace=True)

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

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

```
/usr/local/lib/python3.6/site-packages/pandas/core/generic.py:3110: 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.htm
  self[name] = value
/usr/local/lib/python3.6/site-packages/pandas/core/generic.py:3924: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.htm
  self._update_inplace(new_data)
/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.htm
  self.obj[item] = s


Chromosome           float64
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
```

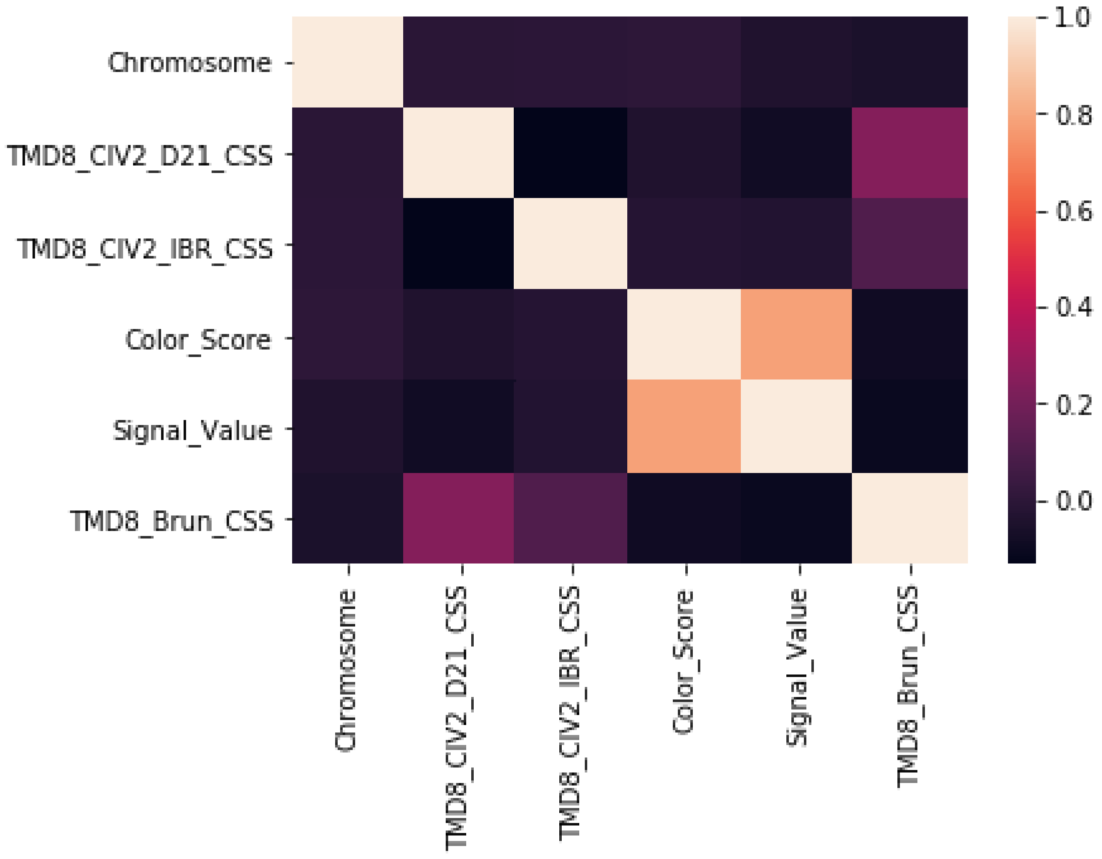
```
Out[67]:    Chromosome  TMD8_CIV2_D21_CSS  TMD8_CIV2_IBR_CSS  Color_Score  \
        0        19.0          -0.731734              -1.15       3249.0
        1        19.0          -0.602970              -1.21       3249.0
        2        19.0          -0.005454              -2.05       3249.0
        3        19.0           0.399748              -0.82       3249.0
        4        19.0           1.168600              -3.56       3249.0

           Signal_Value                 Unique Gene_Symbol  TMD8_Brun_CSS Essential  \
        0      21.24547  chr194079107540791094        AKT2        -0.4339       NaN
        1      21.24547  chr194079116640791185        AKT2        -0.4339       NaN
```

```
2       21.24547   chr194079116740791186        AKT2          -0.4339        NaN
3       21.24547   chr194079130240791321        AKT2          -0.4339        NaN
4       21.24547   chr194079122040791239        AKT2          -0.4339        NaN

   Fig1
0  AKT2
1  AKT2
2  AKT2
3  AKT2
4  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 [45]: import matplotlib.pyplot as plt
         import seaborn as sns
         %matplotlib inline
         sns.heatmap(df2.corr())
```

```
Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x10b3fd2b0>
```



We see now obvious correlations between any of our numeric columns, besides the CRISPR screens showing some slight correlations. Color_Score and Signal_Value seem to be synonymous based on their correlations, so we will just use Signal_Value in further analysis.

I examined ATAC-seq signal and CRISPR score by chromosome, but found no strong patterns (not shown). I will next show the median of CRISPRi in DMSO and CRISPRi in IBR by chromosome.

```
In [61]: print(df2.groupby('Chromosome').TMD8_CIV2_D21_CSS.median())
         df2.groupby('Chromosome').TMD8_CIV2_IBR_CSS.median()
```

```
Chromosome
1.0     0.042811
2.0     0.057783
3.0     0.061304
4.0     0.071740
5.0     0.043502
6.0     0.060810
7.0     0.073865
8.0     0.053018
9.0     0.050779
10.0    0.049296
11.0    0.074679
12.0    0.056998
13.0    0.024557
14.0    0.056498
15.0    0.046867
16.0    0.053590
17.0    0.055071
18.0    0.105353
19.0    0.072458
20.0    0.042488
21.0    0.021125
22.0    0.048240
23.0    0.045828
24.0    0.098778
Name: TMD8_CIV2_D21_CSS, dtype: float64
```

```
Out[61]: Chromosome
         1.0    -0.160
         2.0    -0.170
         3.0    -0.150
         4.0    -0.150
         5.0    -0.170
         6.0    -0.150
         7.0    -0.140
         8.0    -0.150
```

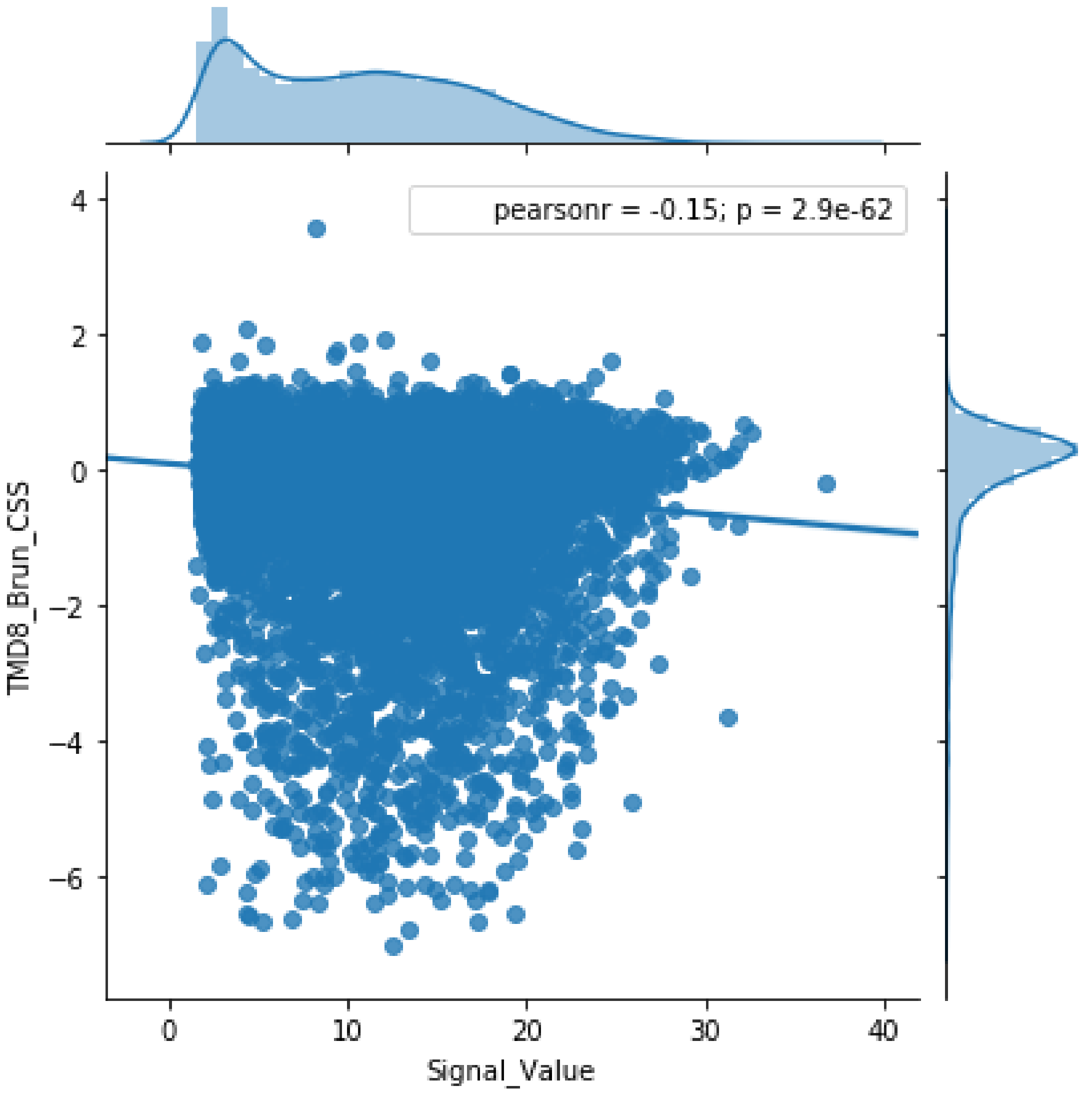
```
9.0     -0.150
10.0    -0.140
11.0    -0.170
12.0    -0.150
13.0    -0.170
14.0    -0.180
15.0    -0.160
16.0    -0.170
17.0    -0.170
18.0    -0.160
19.0    -0.140
20.0    -0.175
21.0    -0.150
22.0    -0.170
23.0    -0.165
24.0    -0.120
Name: TMD8_CIV2_IBR_CSS, dtype: float64
```

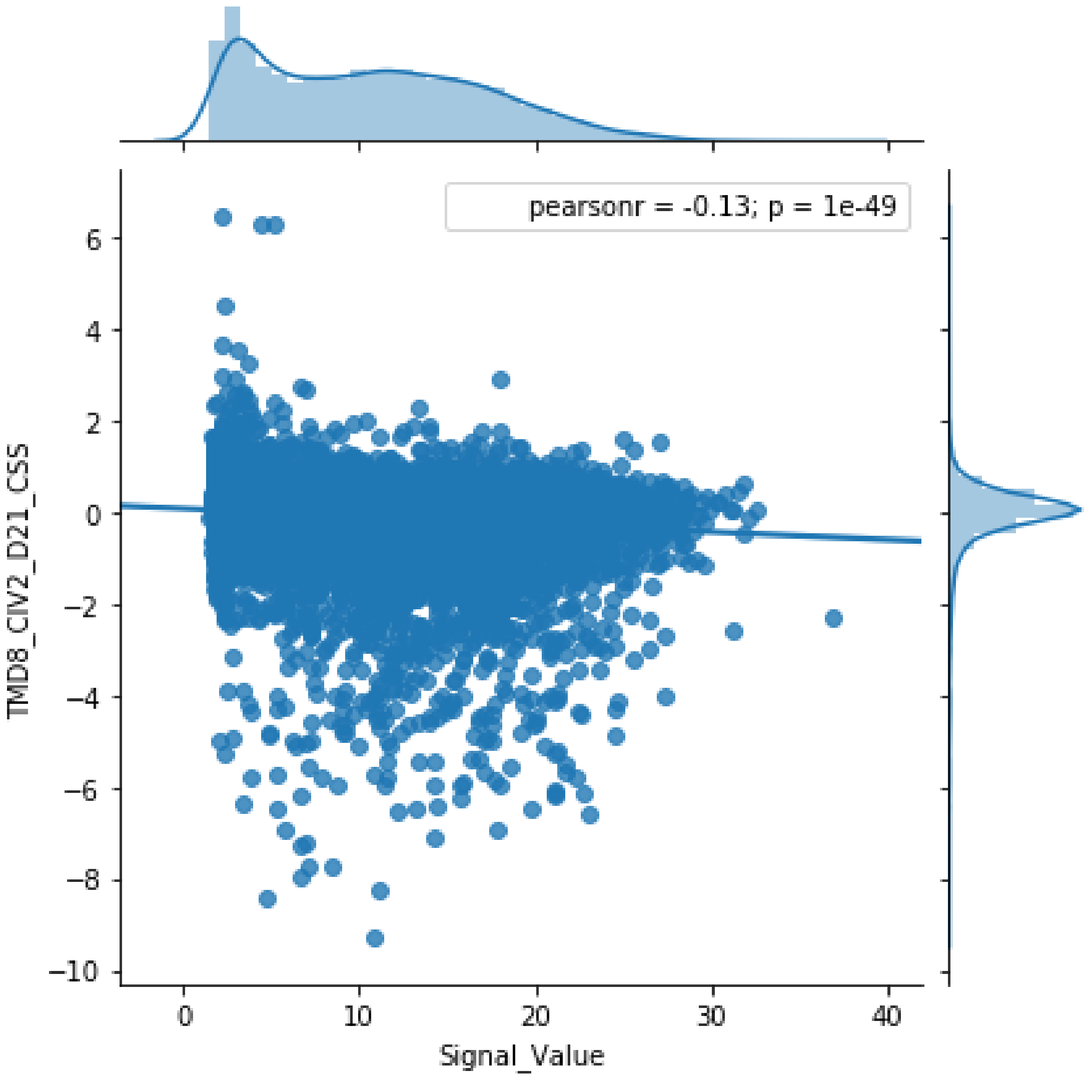
Interestingly, the median of the IBR CRISPRi score is negative for every chromosome, whereas it is not in DMSO

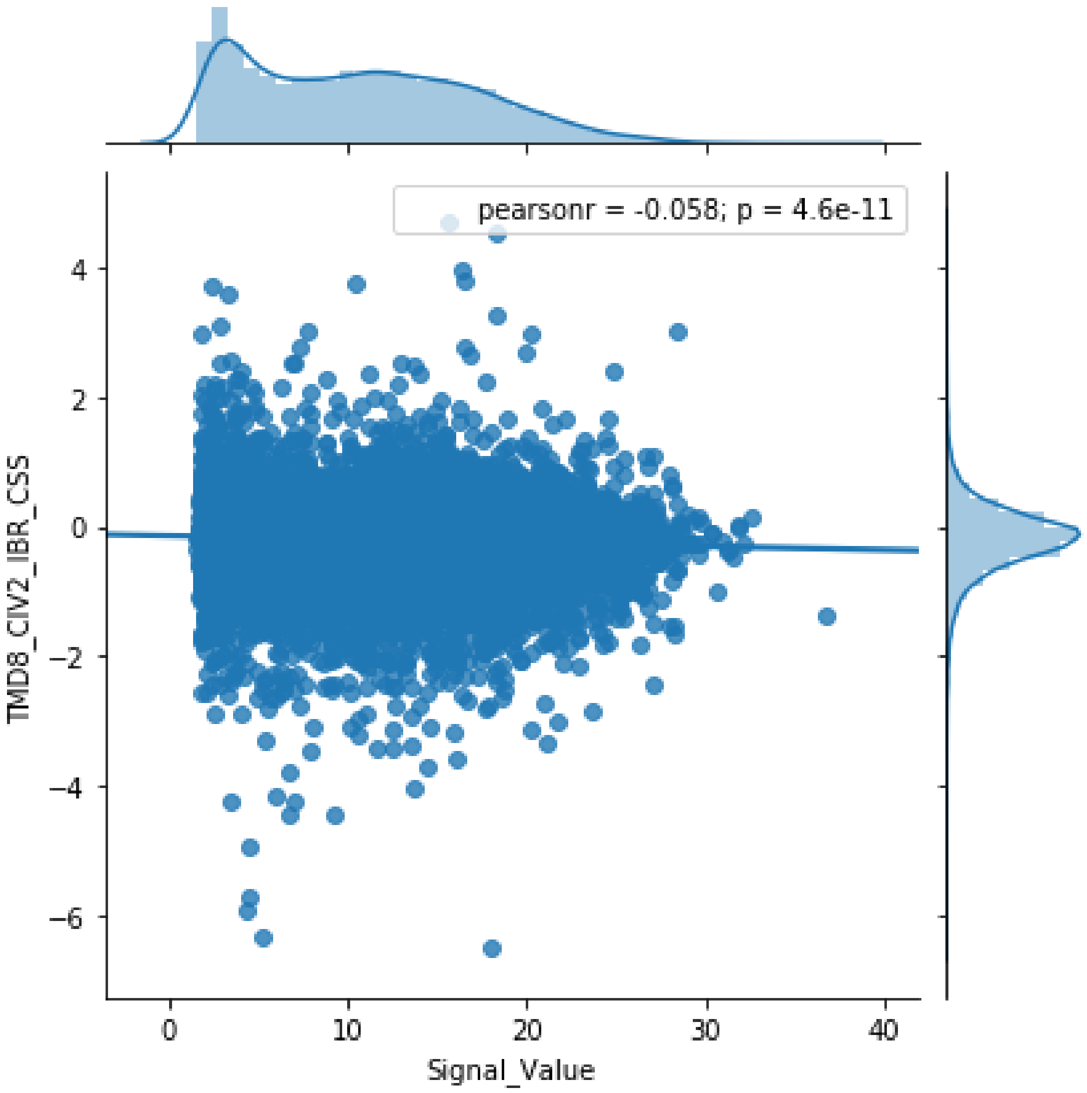
Let's begin to visualize some the data. We are interested in the interplay between ATAC-seq signal and CRISPR CSS. We will start by examining all of the genes for each screen.

```
In [68]: 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[68]: <seaborn.axisgrid.JointGrid at 0x118d0b160>
```


pearsonr = -0.15; p = 2.9e-62
TMD8_Brun_CSS
Signal_Value
4
2
0
−2
−4
−6
0
10
20
30
40

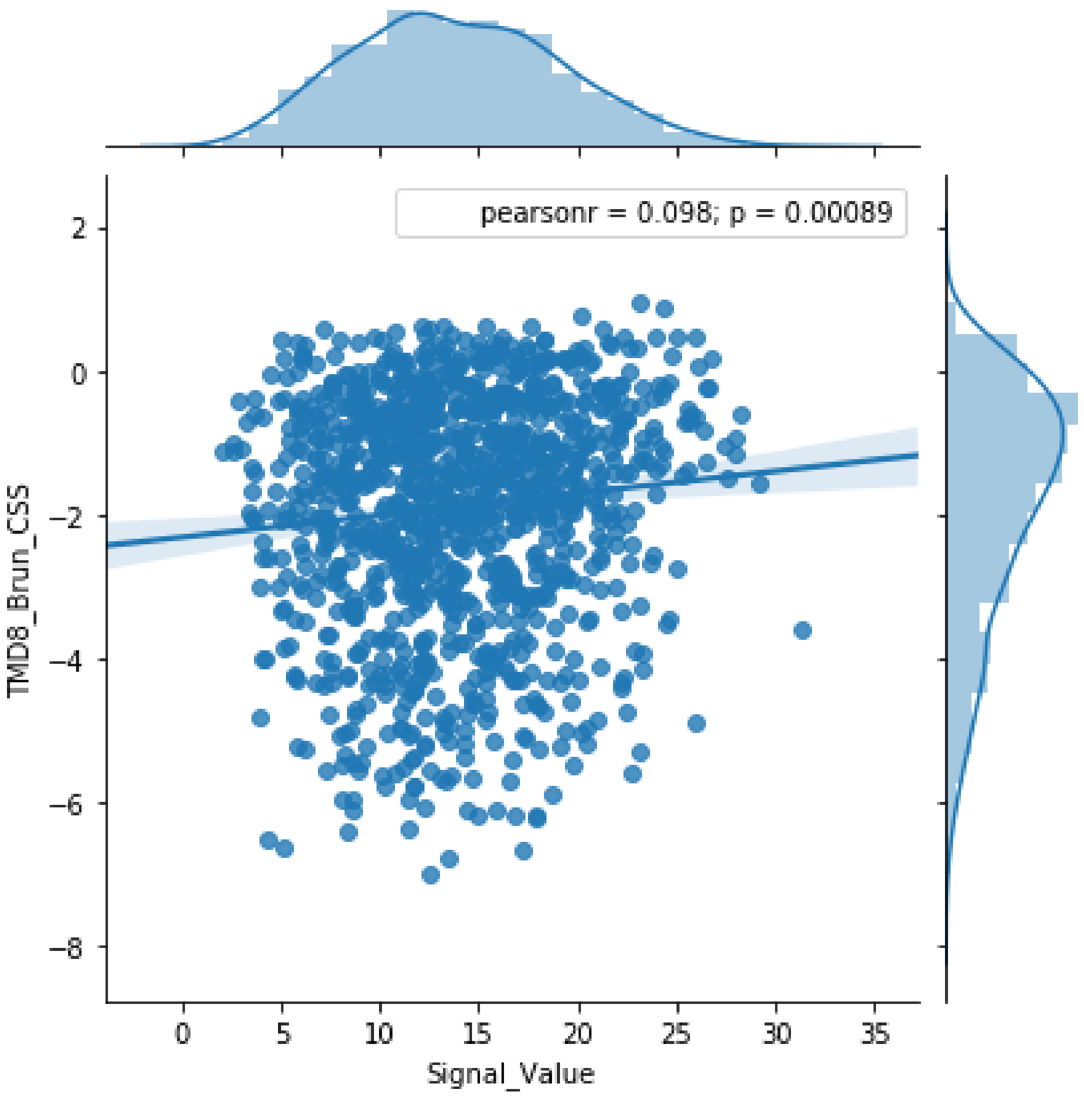

pearsonr = -0.13; p = 1e-49
TMD8_CIV2_D21_CSS
Signal_Value
6
4
2
0
−2
−4
−6
−8
−10
0
10
20
30
40



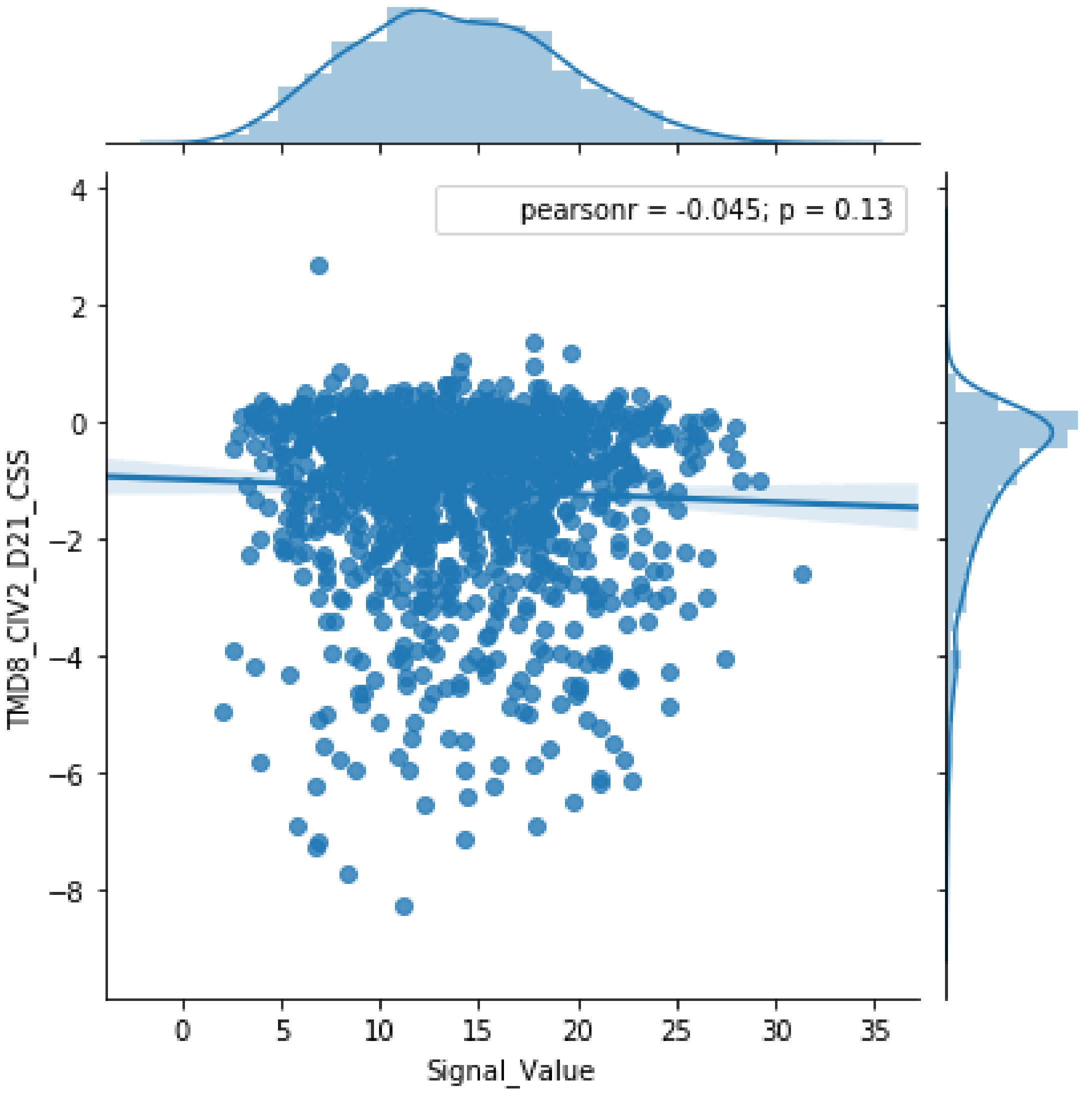
There is no easily discernible trend between our CRISPR screening scores and ATAC data when looking at every gene. Let's try subsetting the data. First, we will look at "Essential" genes. These are genes published by the Hart lab that have been deterimined to be essential for every cell line.

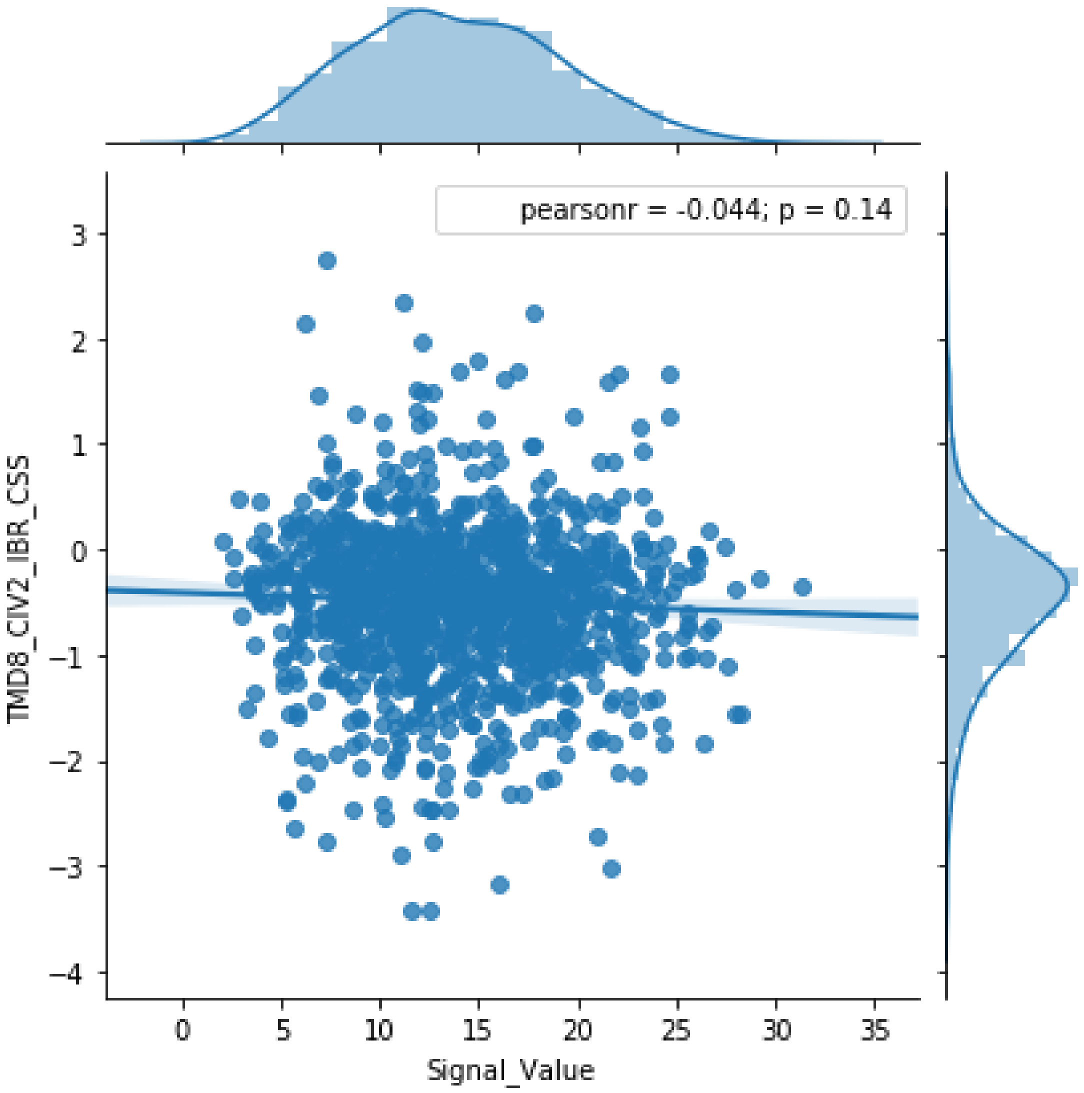
```
In [69]: sns.jointplot(x = 'Signal_Value', y = 'TMD8_Brun_CSS', data = df2.dropna\
          (subset = ['Signal_Value', 'Essential']).groupby('Gene_Symbol').mean(), kind = 'reg'

         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[69]: <seaborn.axisgrid.JointGrid at 0x10ec2dcf8>
```



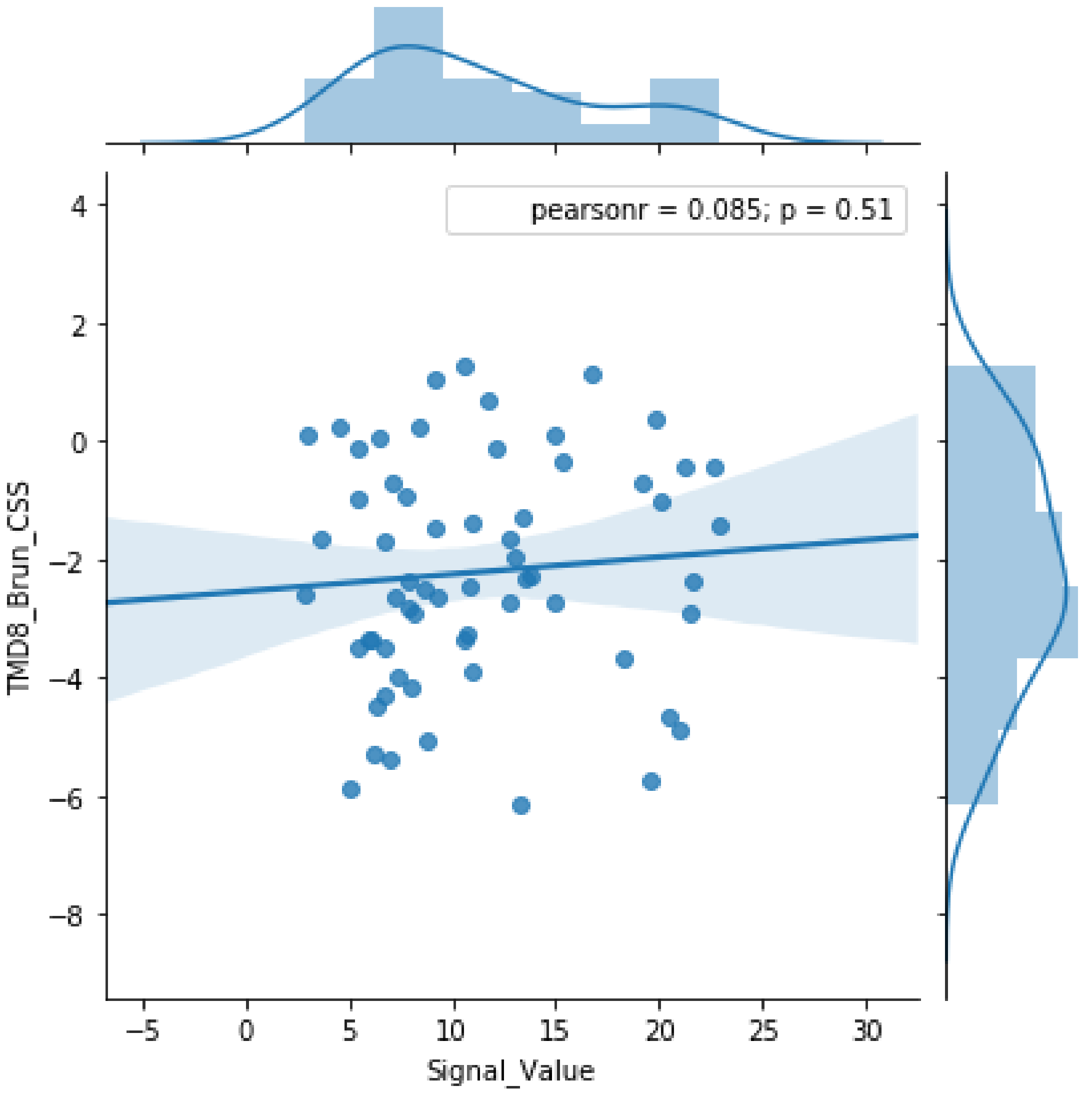
pearsonr = 0.098; p = 0.00089
2
0
−2
−4
−6
−8
TMD8_Brun_CSS
0
5
10
15
20
25
30
35
Signal_Value

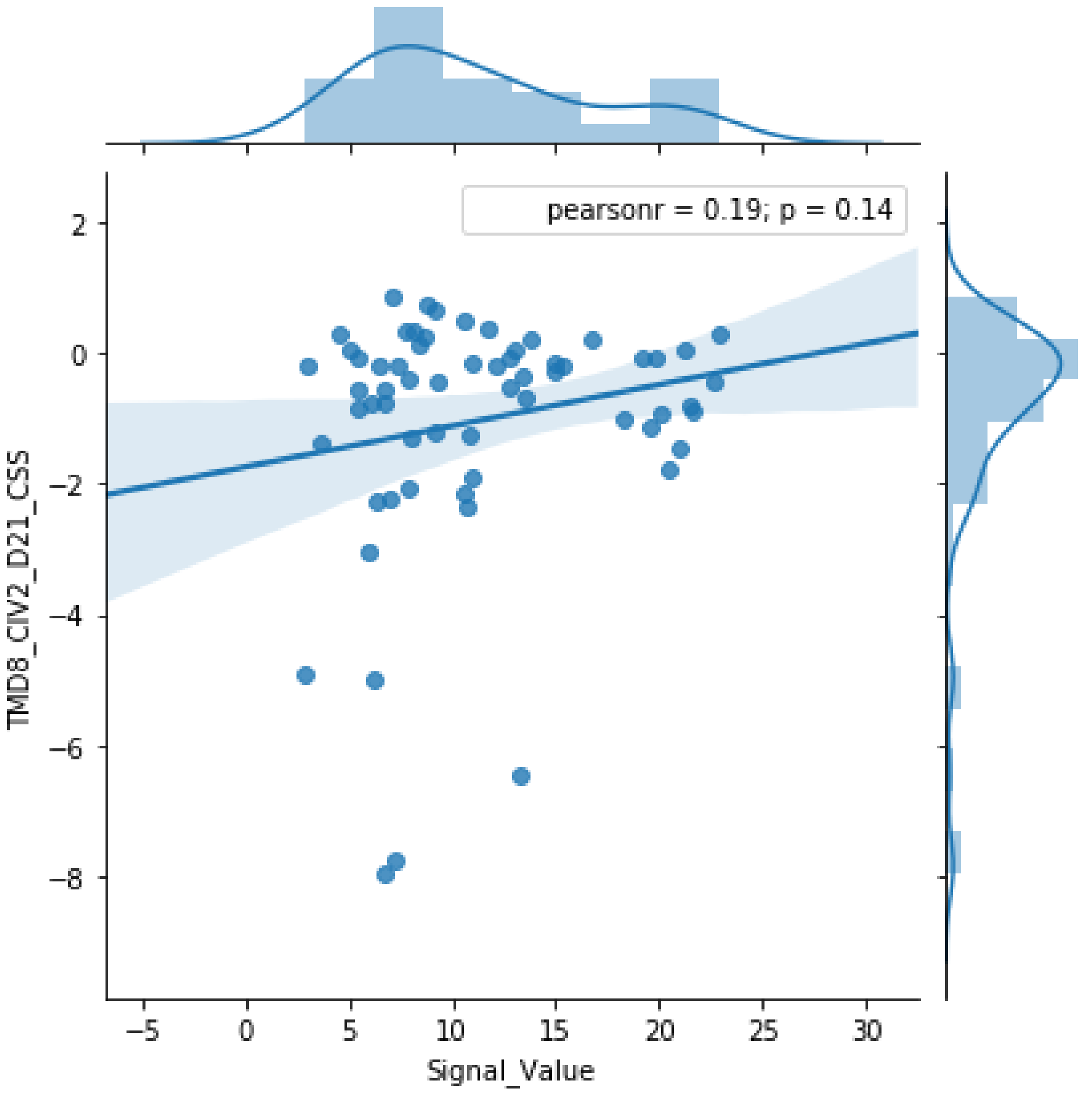

pearsonr = -0.045; p = 0.13
TMD8_CIV2_D21_CSS
Signal_Value
4
2
0
−2
−4
−6
−8
0
5
10
15
20
25
30
35

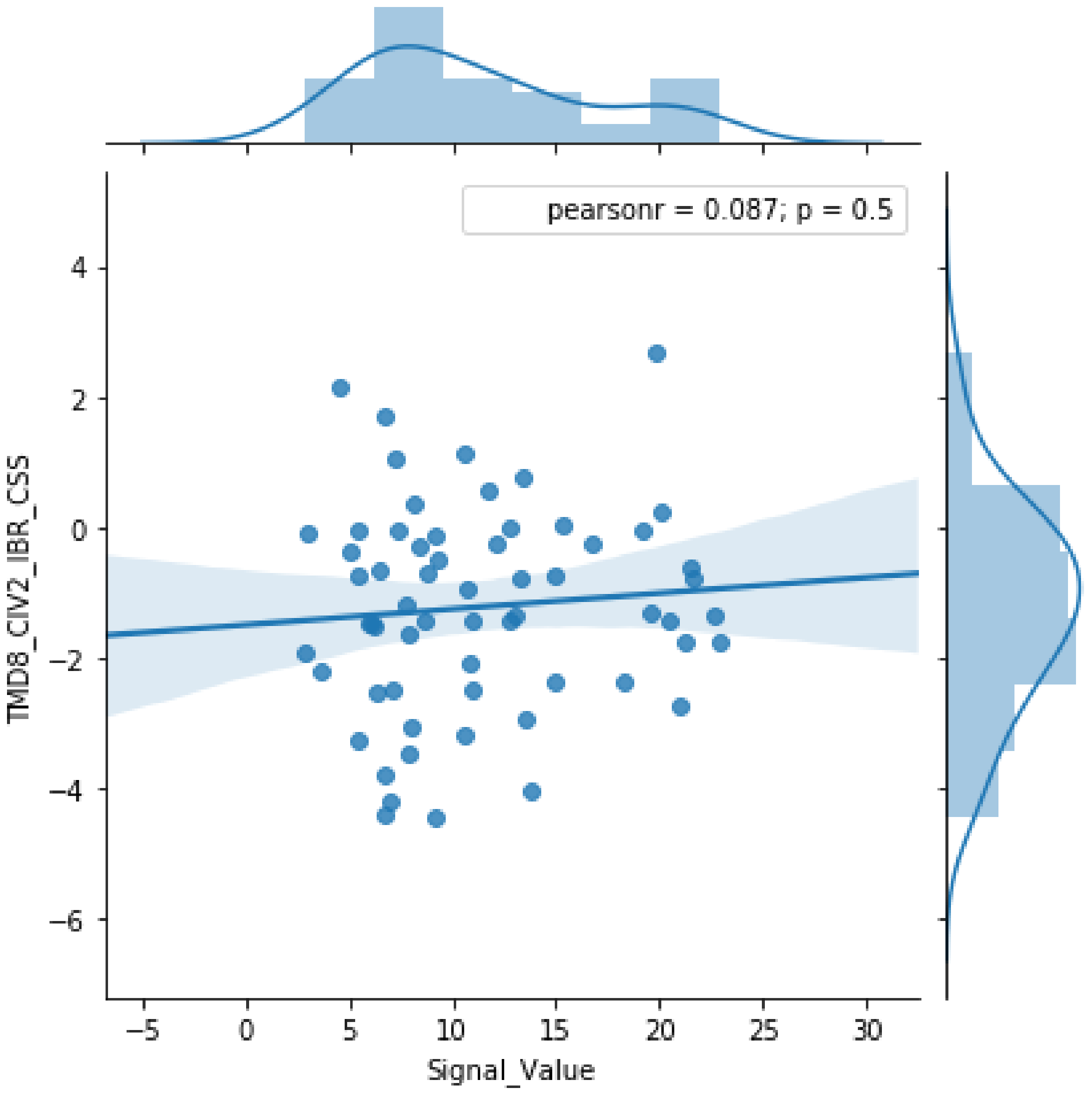


No interesting conclusions arise from this. Let's now try this with "Figure 1" genes. These are genes that are on average essential to most ABC DLBCL cell lines.

```
In [70]: 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[70]: <seaborn.axisgrid.JointGrid at 0x114df2518>
```


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

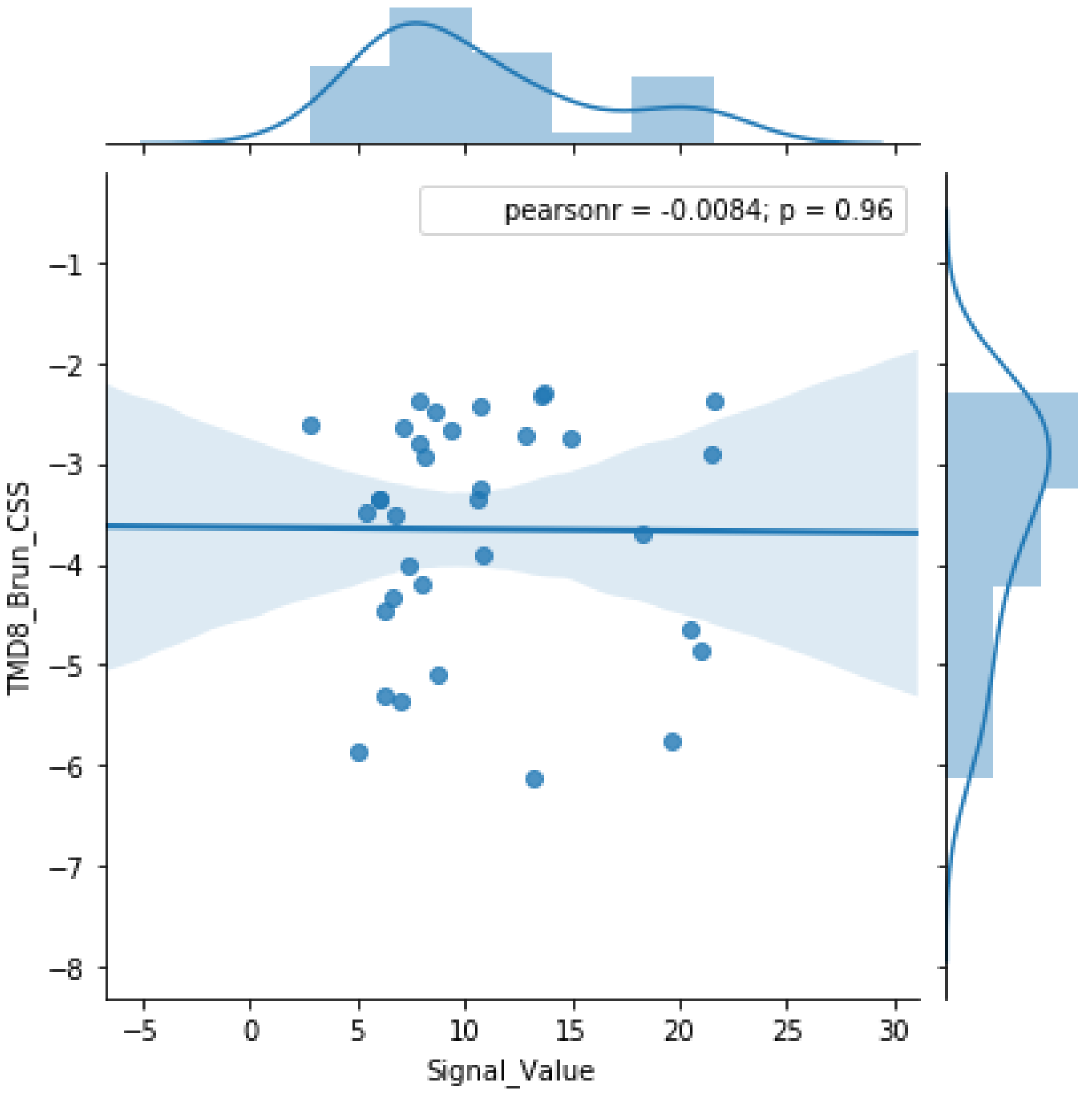

pearsonr = 0.19; p = 0.14
TMD8_CIV2_D21_CSS
Signal_Value
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25
30

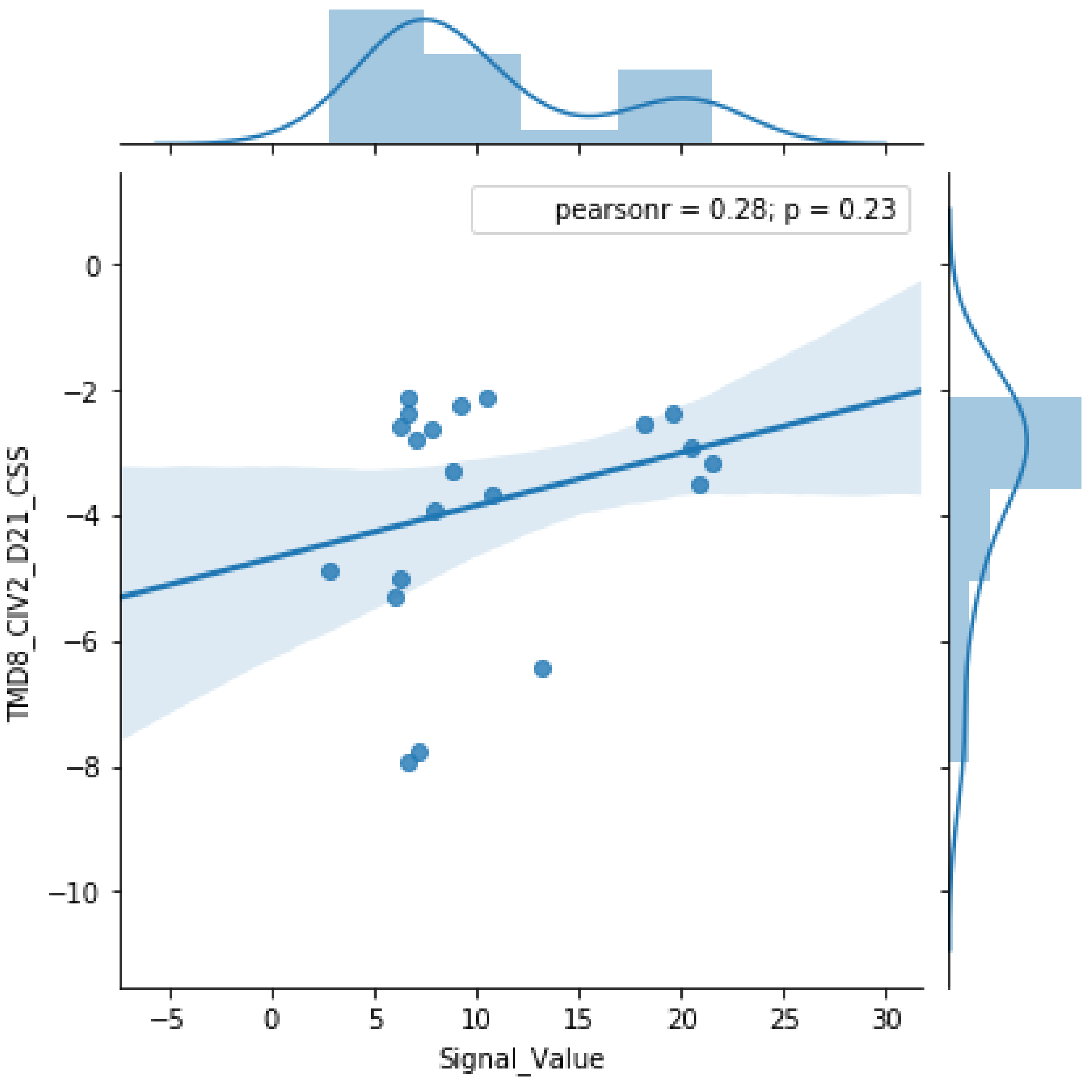


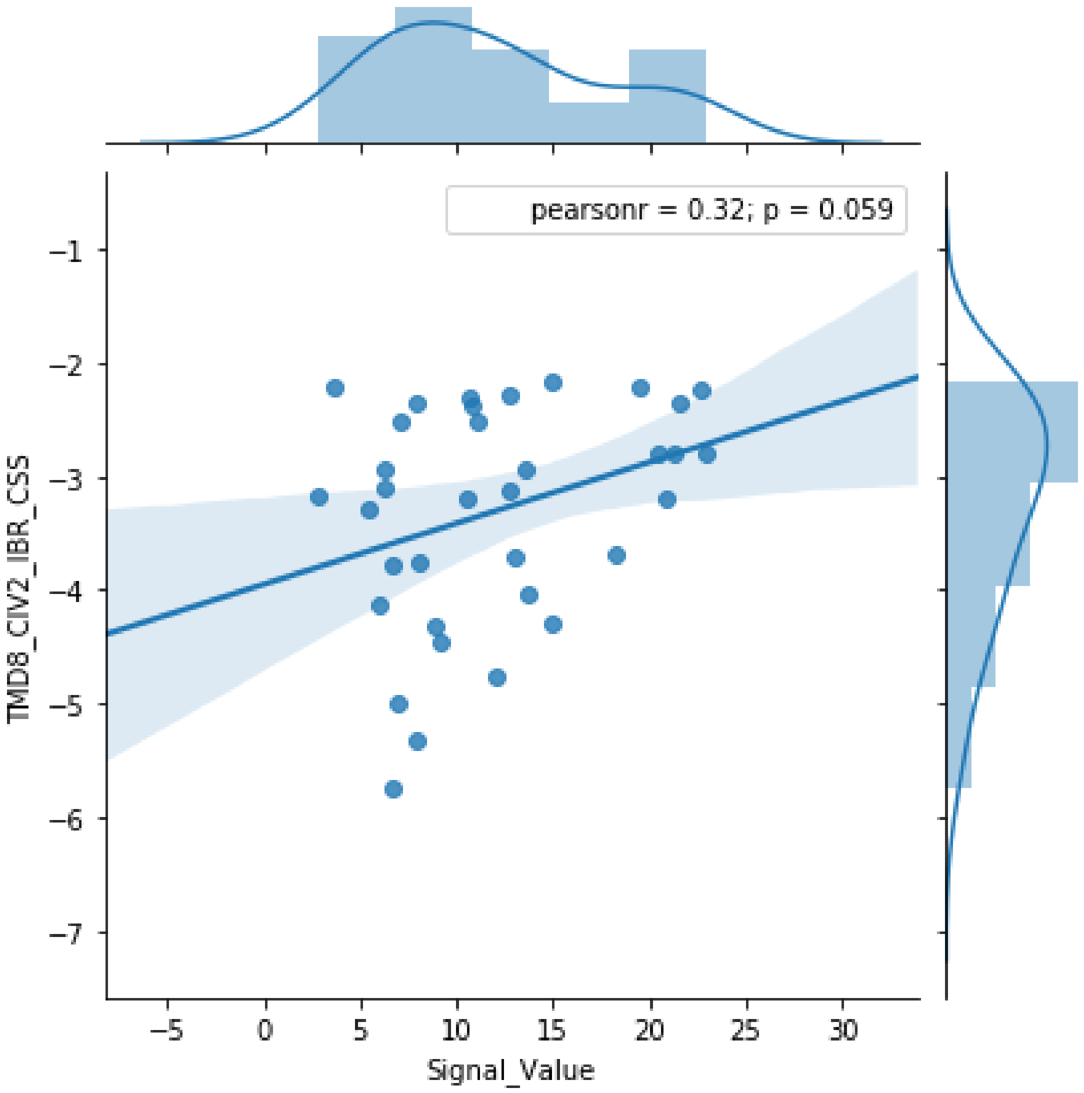
Let's now look at the figure 1 genes where the CRISPR score was < -2. This will be a list of genes that have been deterimined on average to be essential and did well in each of the screens.

```
In [8]: 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[8]: <seaborn.axisgrid.JointGrid at 0x107d456a0>
```




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



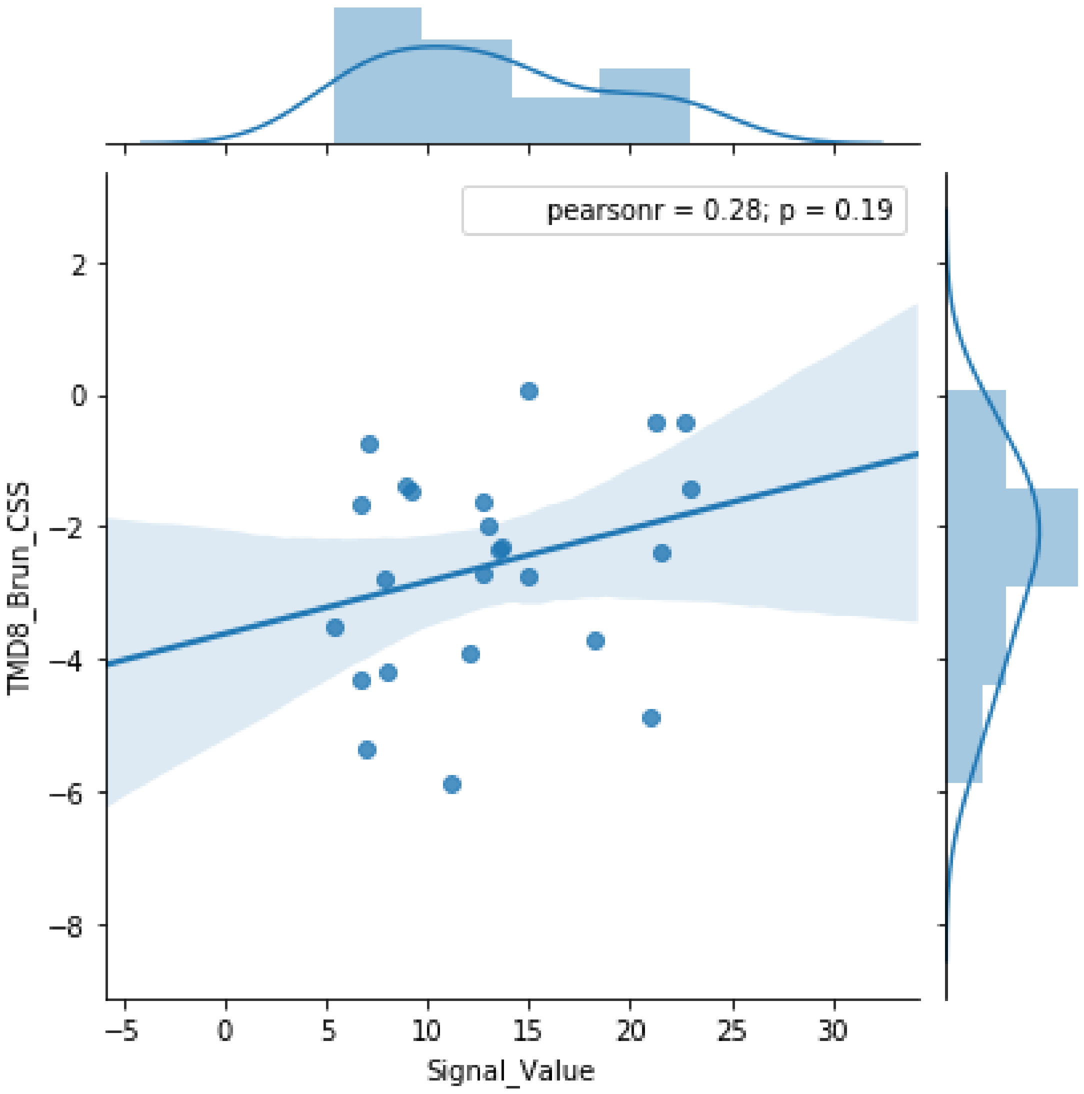
Now we have something that may be interesting. We can see here that there is no correlation between ATAC signal and CRISPR cutting score. However, we do see a slight correlation between ATAC signal CRISPRi score in both DMSO and IBR. There is an direct relationship here - the lower the ATAC signal, the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

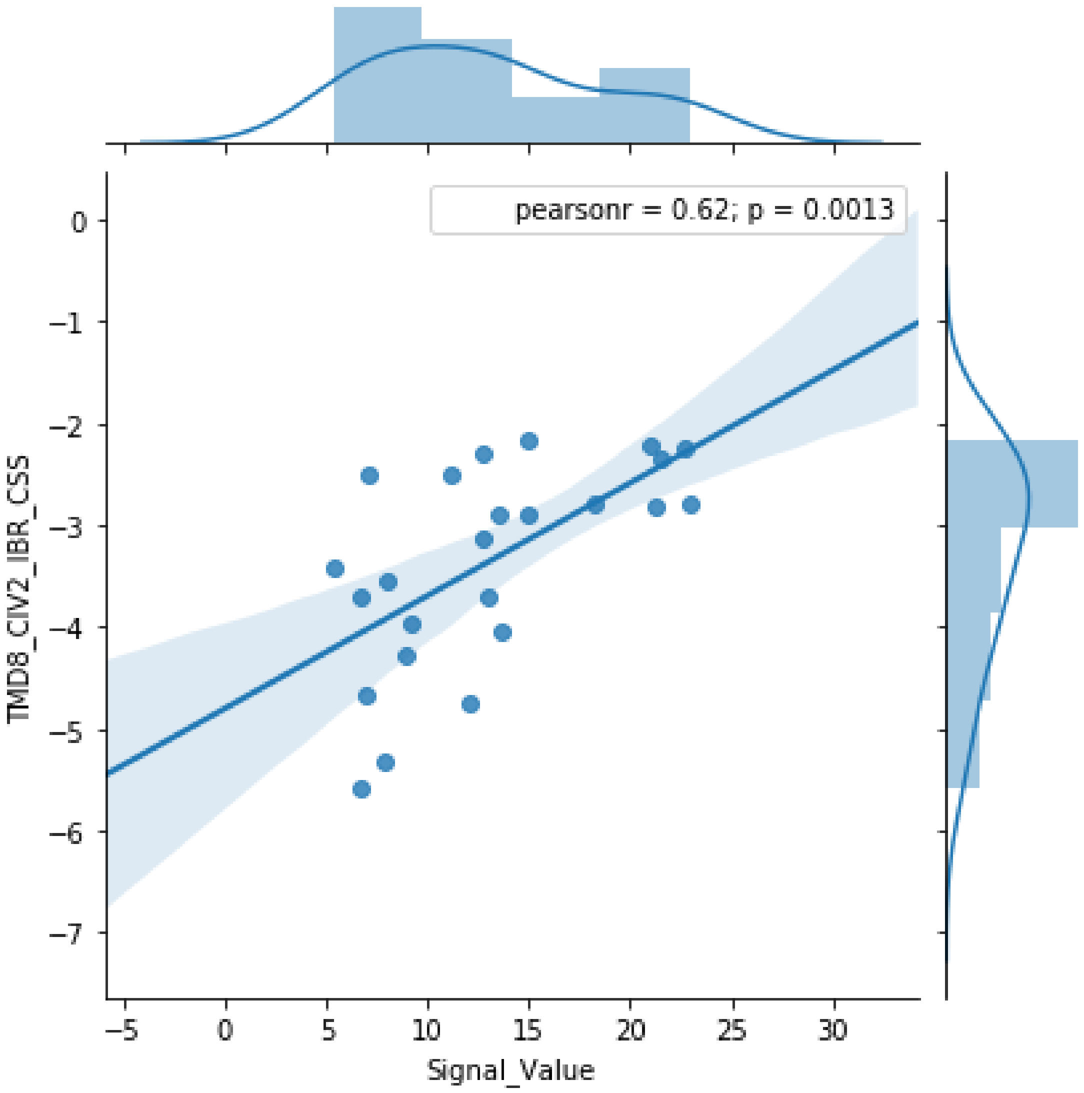
Let's look at genes from figure 1 that performed well in IBR, but did not perform well in DMSO. This will be genes that had a CSS < -2 in IBR and > -1 in DMSO.

```
In [71]: sns.jointplot(x = 'Signal_Value', y = 'TMD8_Brun_CSS', data = df2\
         [(df2.TMD8_CIV2_IBR_CSS < -2) & (df2.TMD8_CIV2_D21_CSS > -1)].dropna(subset = ['Signa
         ('Gene_Symbol').mean(), kind = 'reg')
```

```
sns.jointplot(x = 'Signal_Value', y = 'TMD8_CIV2_IBR_CSS', data = df2\
[(df2.TMD8_CIV2_IBR_CSS < -2) & (df2.TMD8_CIV2_D21_CSS > -1)].dropna(subset = ['Signa
('Gene_Symbol').mean(), kind = 'reg')
```

```
Out[71]: <seaborn.axisgrid.JointGrid at 0x11aa7dbe0>
```





Now, we seen a pretty good correlation between the CRISPRi CSS in IBR and ATAC-seq signal value with a pearson r value of 0.62.

Where do we go from here?

Perhaps IBR lowers the amount of basal transcription of these figure 1 genes. We know that IBR decreases NF-kB, so maybe NF-kB responsive genes are expressed less in cells treated with IBR, allowing CRISPRi to truly silence genes that it was unable to in DMSO. We should compare CSS and ATAC-seq signal on NF-kB responsive genes next.