Try-out with all kinds of statistics on harder dataset

(upto July 3 2018)

In [3]:
```
import numpy as np
import imgutils
import matplotlib.pyplot as plt
```

In [225]:
```
# Re-run this cell if you altered imgutils or imgutils_test
import importlib
importlib.reload(imgutils)
```

Out[225]:
```
<module 'imgutils' from 'C:\\JADS\\SW\\Grad Proj\\realxtals1\\sources\\imgutils.py'>
```

In [150]:
```
df_imgfiles = imgutils.scanimgdir('../data/Crystals_Apr_12/Tileset6_subset', '.tif')
img = imgutils.loadtiff(df_imgfiles['filename'][1])
```

In [151]:
```
print(np.min(img))
```

```
180
```

In [152]:
```
a = np.histogram(img, 8192)
```

In [ ]:
```

```

In [153]:
```
def mvalue(hist):
    sum = 0
    prevcount = hist[0]
    for index, count in np.ndenumerate(hist):
        if (index == 0): continue
        sum = sum + np.abs(count - prevcount)
        prevcount = count
    mvalue = sum / np.max(hist)
    return mvalue
```

In [154]:
```
m = imgutils.img_shapevalue(img)
print(m)
```

```
1.9815967456685324
```

In [155]:
```
hist = np.histogram(img, 1024)[0]
plt.plot(hist)
```

Out[155]:
```
[<matplotlib.lines.Line2D at 0xf78c0b8>]
```

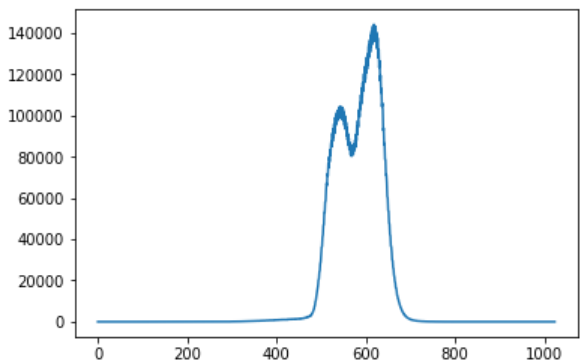
In [156]:
```
img2 = img[256:512,0:100]
```

In [157]:
```
m2 = imgutils.img_shapevalue(img2)
print(m2)
```

```
1.9807391206570466
```

In [250]:
```
hist2 = imgutils.img_histogram(img, 1024)[0]
plt.plot(hist2)
```

Out[250]:
```
[<matplotlib.lines.Line2D at 0x108f20b8>]
```

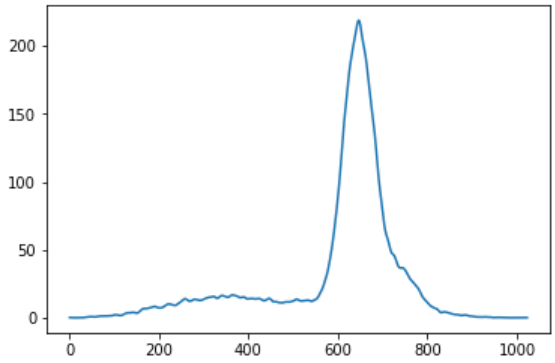


In [251]:
```
def smooth(hist, window=5):
    sigma = window
    gaussian_func = lambda x, sigma: 1/np.sqrt(2*np.pi*sigma**2) * np.exp(-(x**2)/(2*sigma**2))
    gau_x = np.linspace(-2.7*sigma, 2.7*sigma, 6*sigma)
    gau_mask = gaussian_func(gau_x, sigma)
    return np.convolve(hist, gau_mask, 'same')
```

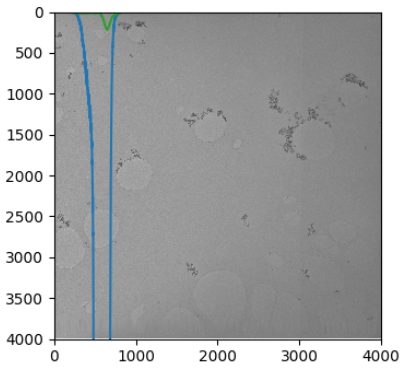
In [252]:
```
hist2 = np.histogram(img2, bins=1024)[0]
hist3 = imgutils.smooth_histogram(hist2,4)
m3 = imgutils.img_shapevalue(img2)
print(m3)
plt.plot(hist3)
```

```
1.9807391206570466
```

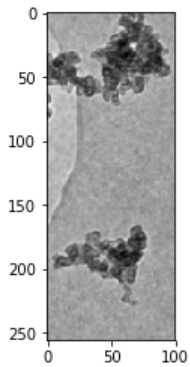
Out[252]:
```
[<matplotlib.lines.Line2D at 0x117eef28>]
```



In [161]:
```
imgutils.showimg(img)
```



In [16]:
```
imgutils.showimg(img2)
```



In [162]:
```
print(np.min(img2))
```

```
5668
```

In [163]:
```
h = imgutils.img_histogram(img2, 65536, normalize=False)
```

In [253]:
```
mean2 =imgutils.img_mean(img2)
low2 = imgutils.img_blacktail(img2, 0.001)
high2 = imgutils.img_whitetail(img2, 0.001)
print(mean2, low2, high2)
```

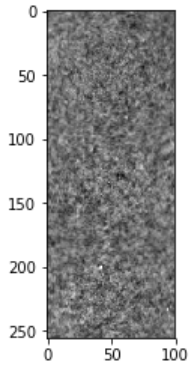
```
12229.6170703125 6390.396000000001 15601.015000000025
```

In [165]:
```
h3 = np.histogram(img2, 8, range=(low2,high2), density=False)
h4 = h3[0] # smooth(h3[0],3)
print(mvalue(h4))
plt.plot(h4)
print(imgutils.img_quartile1(img2), imgutils.img_quartile2(img2), imgutils.img_quartile3(img2))
```

```
1.9856614835585011
12168.0 12642.0 13018.0
```

plt.plot(h[0])

In [254]:
```
img3 = img[256:512,100:200]
imgutils.showimg(img3)
print(imgutils.img_quartile1(img3), imgutils.img_quartile2(img3), imgutils.img_quartile3(img3))
```




```
12312.0 12518.0 12736.0
```

In [167]:
```
h3 = np.histogram(img3, 10, density=False)
h4 = h3[0] # smooth(h3[0],32)
print(mvalue(h4))
print(imgutils.img_shapevalue(img3))
plt.plot(h4)
```

```
1.9986990893625538
1.9315050570487544
```

Out[167]:
```
[<matplotlib.lines.Line2D at 0x105dc9b0>]
```

In [169]:
```
h = imgutils.img_histogram(img)[0]
plt.plot(h)
```

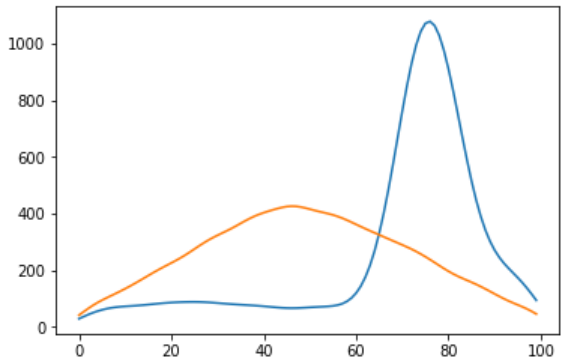
Out[169]:
```
[<matplotlib.lines.Line2D at 0x10a23358>]
```

In [170]:
```
h2 = imgutils.img_histogram(img, bins=100, ref_interval_only=True)[0]
h3 = imgutils.smooth_histogram(h2, 3)
plt.plot(h3)
```

Out[170]:
```
[<matplotlib.lines.Line2D at 0x10a23f60>]
```

In [25]:
```
img2 = img[256:512,0:100]
img3 = img[256:512,100:200]
h2 = imgutils.img_histogram(img2, bins=100, ref_interval_only=True)
h2 = imgutils.smooth_histogram(h2, 3)
h3 = imgutils.img_histogram(img3, bins=100, ref_interval_only=True)
h3 = imgutils.smooth_histogram(h3, 3)
plt.plot(h2)
plt.plot(h3)
print(imgutils.img_shapevalue(img2, bins=100, smoothing_window=5, dynamic_range_only=True))
print(imgutils.img_shapevalue(img3, bins=100, smoothing_window=5, dynamic_range_only=True))
```

```
1.9807391206570466
1.9315050570487544
```



In [171]:
```
h2 = imgutils.img_histogram(img2, bins=5, ref_interval_only=True)
```

In [172]:
```
print(h2[0])
```

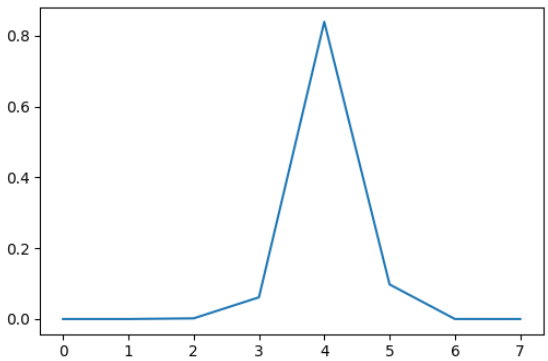
```
[ 1355  1579  1326 13451  6609]
```

In [177]:
```
def hist_bin(img, binnr, nrbins=8):
    return imgutils.img_histogram(img, bins=nrbins, ref_interval_only=False, normalize=True)[0][binnr]

def hist0(img): return hist_bin(img,0)
def hist1(img): return hist_bin(img,1)
def hist2(img): return hist_bin(img,2)
def hist3(img): return hist_bin(img,3)
def hist4(img): return hist_bin(img,4)
def hist5(img): return hist_bin(img,5)
def hist6(img): return hist_bin(img,6)
def hist7(img): return hist_bin(img,7)

def hist_signature():
    return [hist0, hist1, hist2, hist3, hist4, hist5, hist6, hist7]
```

In [178]:
```
a = imgutils.img_histogram(img, bins=8, ref_interval_only=False, normalize=True)
plt.plot(a[0])
plt.show()
```



In [175]:
```
print(df_imgfiles)
```

```
                                            filename
0  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
1  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
2  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
3  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
4  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
5  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
6  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
7  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
8  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
9  ..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...
```

In [179]:
```
stats = hist_signature()
df = imgutils.slicestats(list(df_imgfiles['filename'][:]), 8, 8, stats)
```

In [180]:
```
df.head()
```

Out[180]:

	filename	s_y	s_x	n_y	n_x	alias	hist0	hist1	hist2	hist3	hist4	hist5	hist6	
0	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	0	8	8	img0_0-0	0.010884	0.303728	0.533848	0.140496	0.010676	0.000352	0.000012	0.(
1	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	1	8	8	img0_0-1	0.003476	0.135024	0.453784	0.318208	0.078920	0.009812	0.000756	0.(
2	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	2	8	8	img0_0-2	0.636900	0.265836	0.064644	0.019304	0.010200	0.002692	0.000356	0.(
3	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	3	8	8	img0_0-3	0.251680	0.066332	0.030412	0.019468	0.011924	0.013984	0.542448	0.(
4	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	4	8	8	img0_0-4	0.042324	0.021740	0.020692	0.014024	0.012132	0.021384	0.836744	0.(

In [183]:
```
stat_names = imgutils.stat_names(stats)
stat_normnames = imgutils.normalized_names(stat_names)
imgutils.normalize(df, stat_names)
```

In [184]:
```
df.head()
```

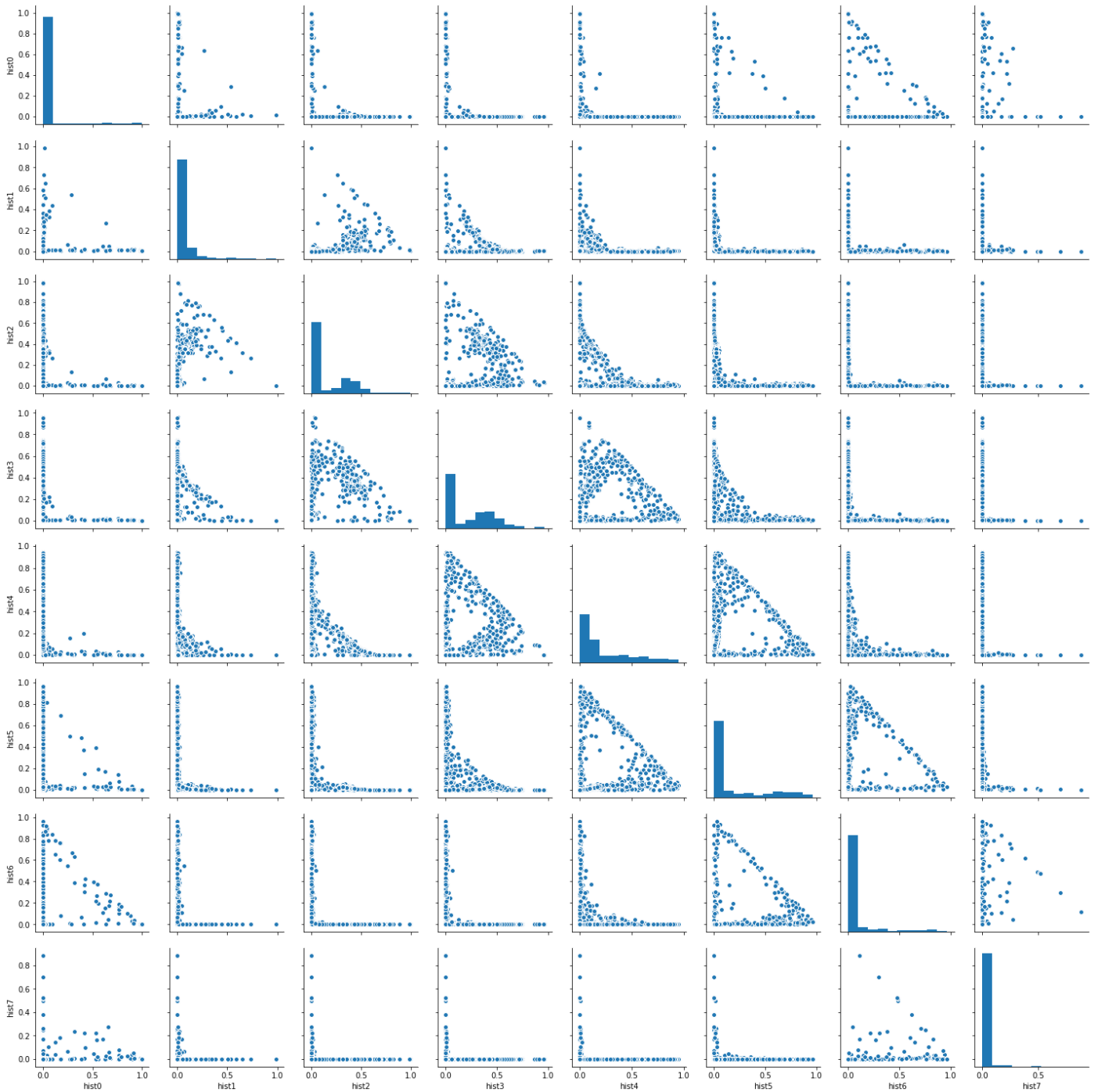
Out[184]:

	filename	s_y	s_x	n_y	n_x	alias	hist0	hist1	hist2	hist3	...	hist6	hist7	|hist0|
0	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	0	8	8	img0_0-0	0.010884	0.303728	0.533848	0.140496	...	0.000012	0.000004	-0.214925
1	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	1	8	8	img0_0-1	0.003476	0.135024	0.453784	0.318208	...	0.000756	0.000020	-0.256637
2	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	2	8	8	img0_0-2	0.636900	0.265836	0.064644	0.019304	...	0.000356	0.000068	3.309952
3	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	3	8	8	img0_0-3	0.251680	0.066332	0.030412	0.019468	...	0.542448	0.063752	1.140913
4	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	4	8	8	img0_0-4	0.042324	0.021740	0.020692	0.014024	...	0.836744	0.030960	-0.037898

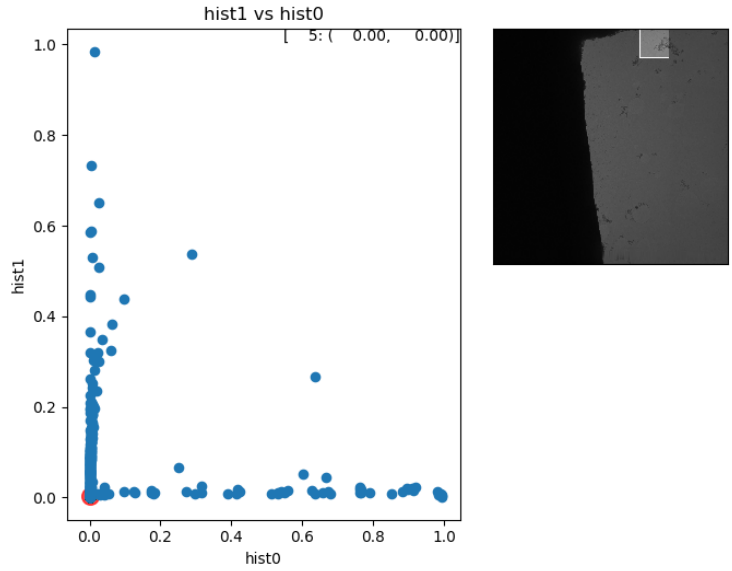
5 rows × 22 columns

In [185]:
```
import seaborn as sb
```

In [186]:
```
%matplotlib inline
sb.pairplot(df, vars=stat_names)
plt.show()
```

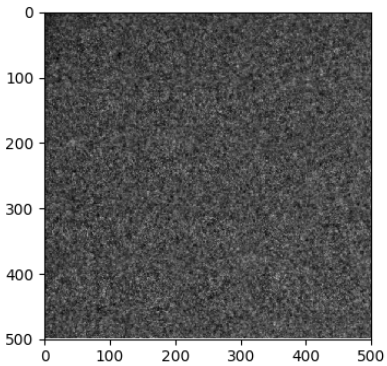


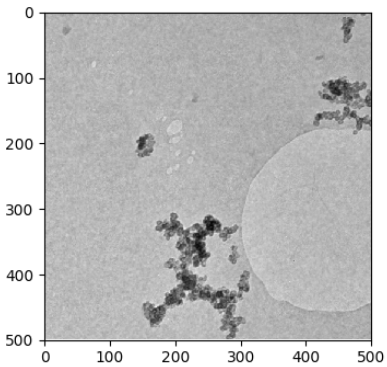
In [188]:
```
%matplotlib notebook
imgutils.plotwithimg(df, 'hist0', 'hist1', imgutils.highlightimgslice, True)
```



In [53]:
```
img0 = imgutils.getimgslice(df,0)
imgN = imgutils.getimgslice(df,20)
```

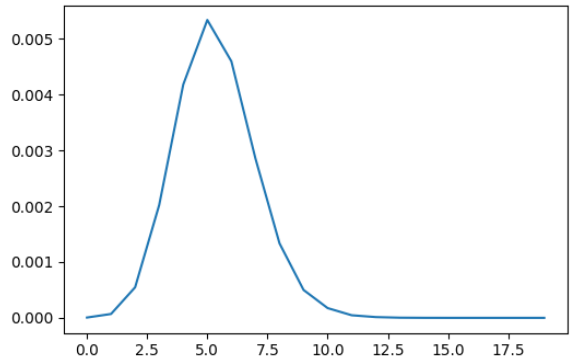
In [54]:
```
imgutils.showimg(img0)
imgutils.showimg(imgN)
```



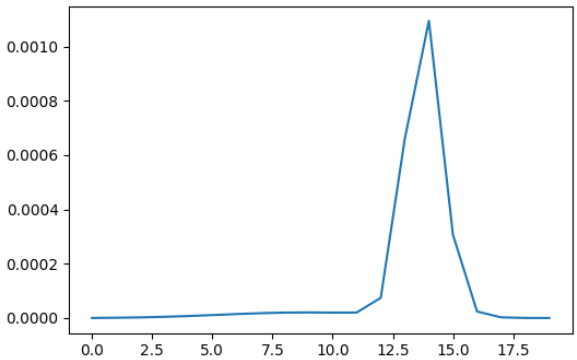


In [55]:
```
a = imgutils.img_histogram(img0, bins=20, normalize=True)
b = imgutils.img_histogram(imgN, bins=20, normalize=True)
```

In [56]:
```
plt.plot(a)
plt.show()
```



In [57]:
```
plt.plot(b)
plt.show()
```



In [43]:
```
print(df[stat_names].iloc[0])
print(df[stat_names].iloc[20])
```

```
hist0    9.443818e-05
hist1    2.635384e-03
hist2    4.632087e-03
hist3    1.219054e-03
hist4    9.263341e-05
hist5    3.054230e-06
hist6    1.041215e-07
hist7    3.470716e-08
Name: 0, dtype: float64
hist0    8.584917e-07
hist1    5.254116e-06
hist2    1.377271e-05
hist3    1.976373e-05
hist4    2.275187e-05
hist5    7.254697e-04
hist6    1.329446e-04
hist7    3.131836e-07
Name: 20, dtype: float64
```

In [58]:
```
import scipy.stats as spstats
```

In [60]:
```
a = spstats.mode(imgN, axis=None)
print(a)
```

```
ModeResult(mode=array([12112], dtype=uint16), count=array([352]))
```

In [61]:
```
print(a[0])
```

```
[12112]
```

In [108]:
```
b = np.histogram(img0, bins=1024, density=False)
imax = np.argmax(b[0])
cmax = np.max(b[0])
emax = b[1][imax]
emaxmin = b[1][imax-1]
gray = emax + (emax - (b[1][imax-1] if (imax>0) else 0)) / 2
print(imax, cmax, emax, emaxmin)
print(gray)
print(spstats.mode(img0, axis=None)[0])
print(b[1][len(b[1])-1])
```

```
293 1402 576.814453125 575.9140625
577.2646484375
[577]
1235.0
```

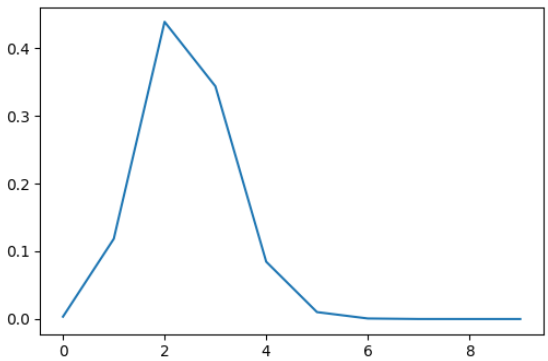
In [127]:
```
c = imgutils.img_mode_cnt(img0, use_fast=True)
print(c)
```

```
1402
```

In [132]:
```
d = imgutils.img_histogram(img0, bins=10, bincenters=False, normalize=True)
e = np.sum(d[0])
print(e)
```

```
1.0
```

In [133]:
```
plt.plot(d[0])
plt.show()
```

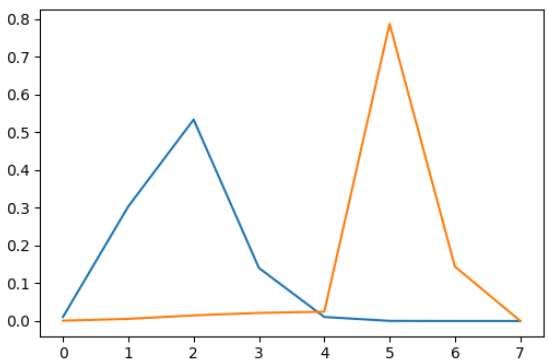


In [200]:
```
# Re-run this cell if you altered imgutils or imgutils_test
import importlib
importlib.reload(imgutils)
```

Out[200]:
```
<module 'imgutils' from 'C:\\JADS\\SW\\Grad Proj\\realxtals1\\sources\\imgutils.py'>
```

In [138]:
```
a0 = imgutils.img_histogram(img0, bins=8, ref_interval_only=False, normalize=True)
aN = imgutils.img_histogram(imgN, bins=8, ref_interval_only=False, normalize=True)

plt.plot(a0[0])
plt.plot(aN[0])
plt.show()
```



In [276]:
```
ats2 = [imgutils.img_mean, imgutils.img_std, imgutils.img_mode, imgutils.img_kurtosis, imgutils.img_skewness, imgutils.img_
2 = imgutils.slicestats(list(df_imgfiles['filename'][:]), 10, 10, stats2)
```

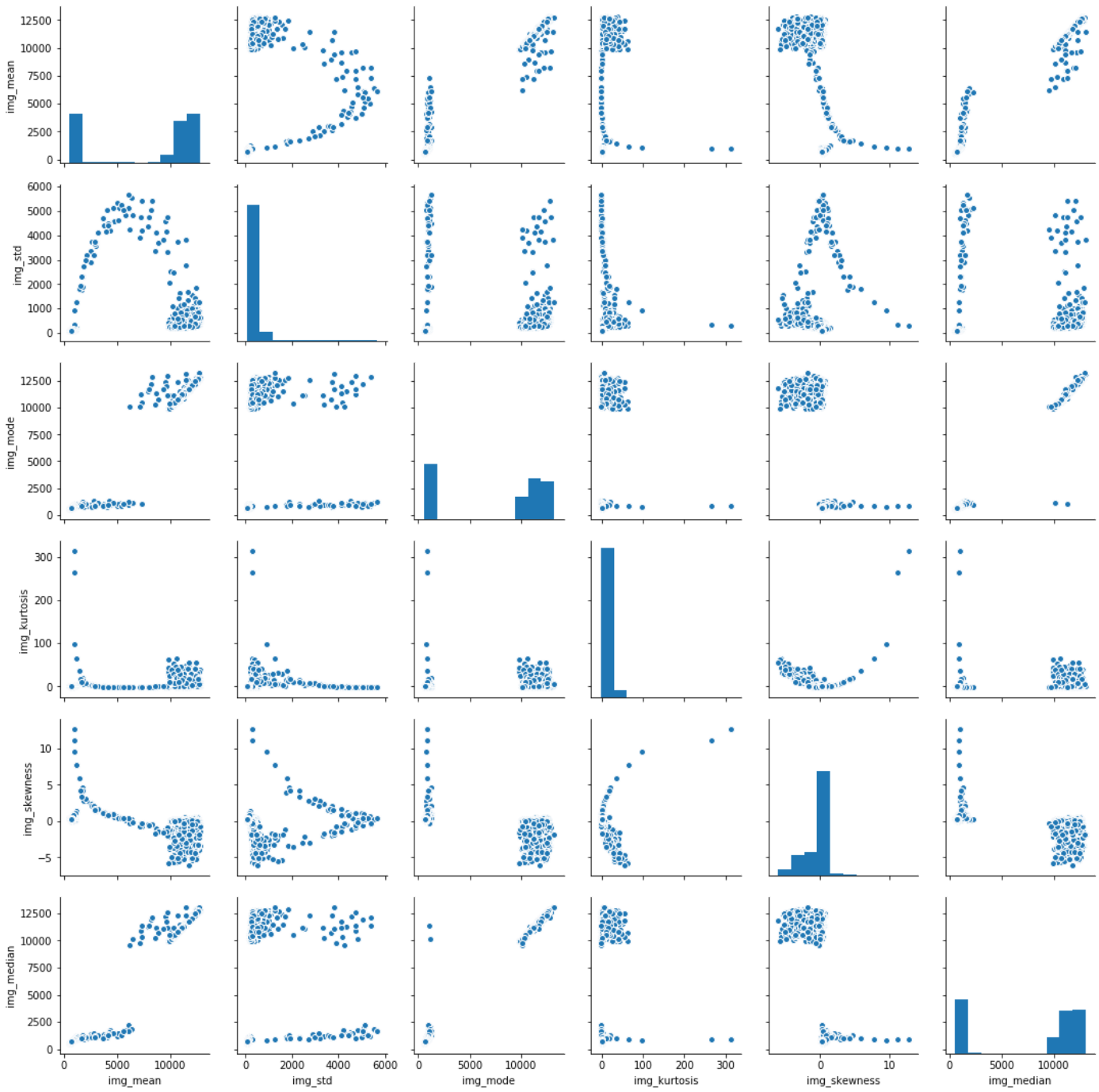
In [277]:
```
df2.head()
```

Out[277]:

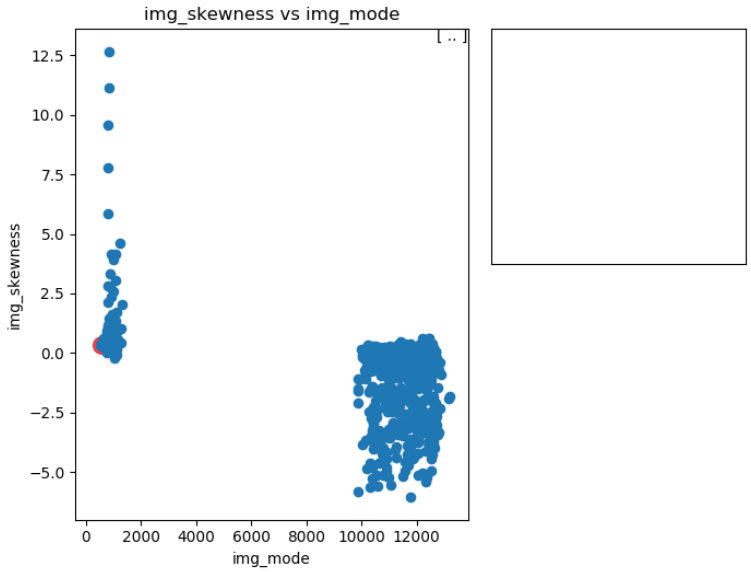
	filename	s_y	s_x	n_y	n_x	alias	img_mean	img_std	img_mode	img_kurtosis	img_skewness	img
0	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	0	10	10	img0_0-0	577.126750	73.520486	567.121582	0.314492	0.357213	
1	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	1	10	10	img0_0-1	597.576300	71.413482	584.238770	0.257486	0.330935	
2	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	2	10	10	img0_0-2	715.021062	117.571635	658.485352	-0.118063	0.402441	
3	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	3	10	10	img0_0-3	2494.602294	2877.999371	993.552246	5.360884	2.566190	
4	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	4	10	10	img0_0-4	7939.682550	4363.425205	11547.012207	-1.722888	-0.445492	

In [278]:
```
stat_names2 = imgutils.stat_names(stats2)
stat_normnames2 = imgutils.normalized_names(stat_names2)
imgutils.normalize(df2, stat_names2)
```

In [279]:
```
%matplotlib inline
sb.pairplot(df2, vars=stat_names2)
plt.show()
```



In [281]:
```
%matplotlib notebook
imgutils.plotwithimg(df2, 'img_mode', 'img_skewness', imgutils.highlightimgslice, True)
```



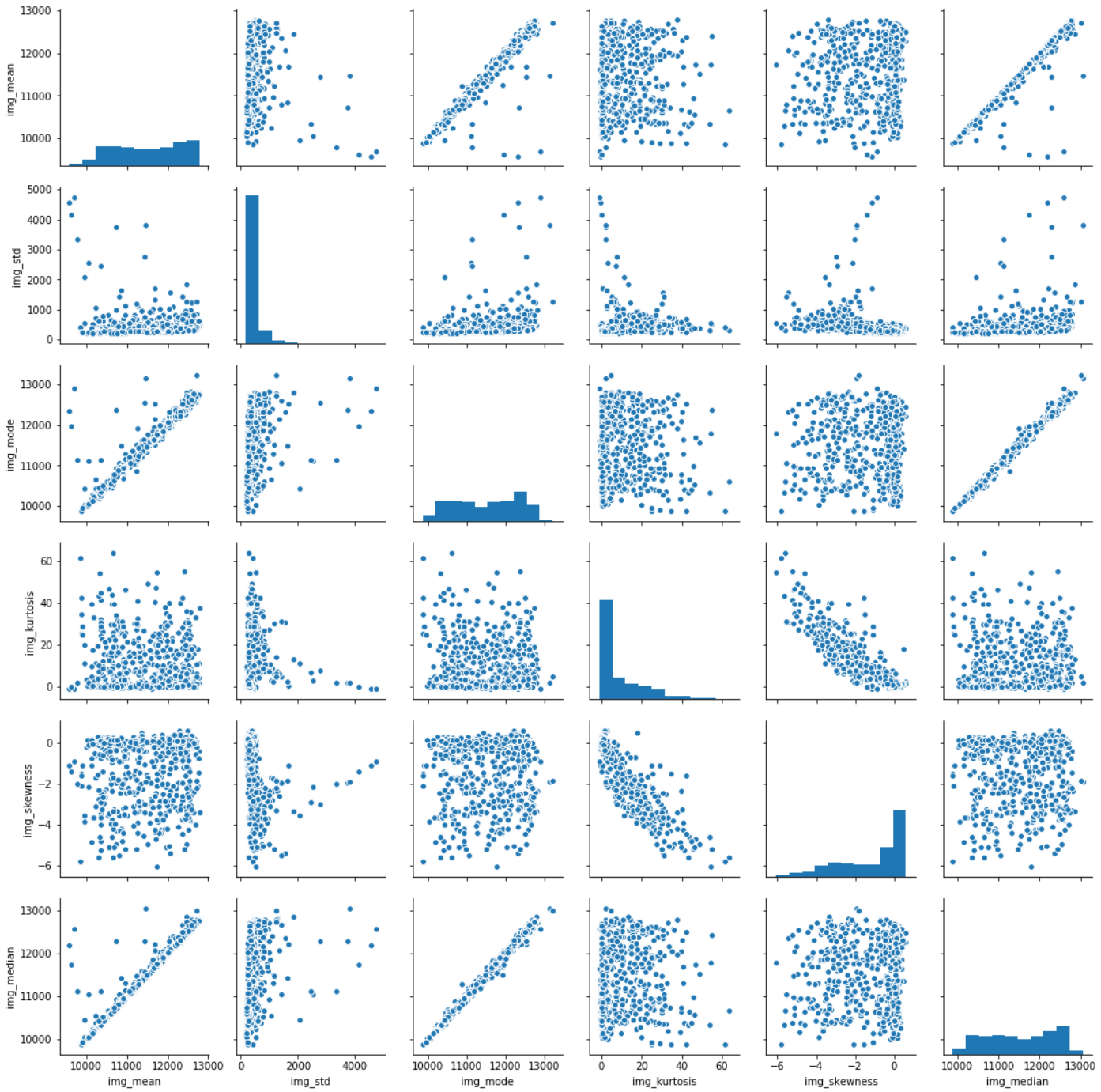
In [282]:
```
df3 = df2[(df2['img_mode']>4000) & (df2['img_mean']>9500)]
```

In [283]:
```
df3.head()
```

Out[283]:

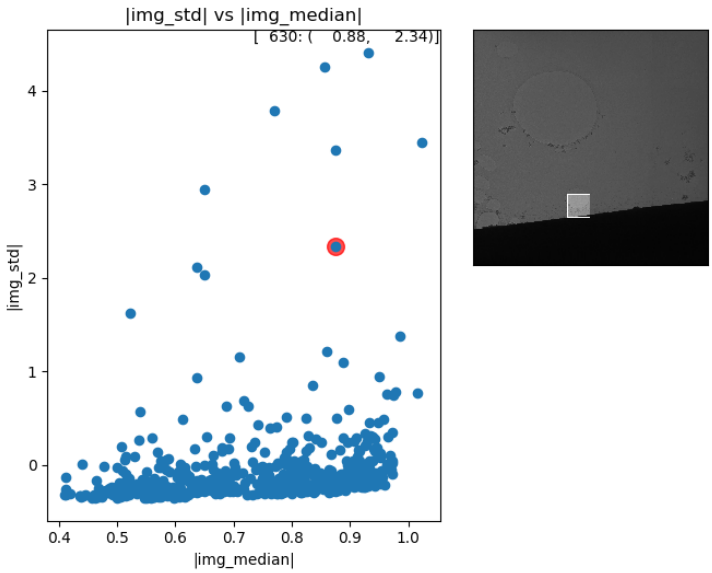
	filename	s_y	s_x	n_y	n_x	alias	img_mean	img_std	img_mode	img_kurtosis	img_skewness	in
5	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	5	10	10	img0_0-5	10046.909687	2537.485858	11106.167969	2.993210	-2.137090	
6	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	6	10	10	img0_0-6	10732.259800	992.587752	10920.350586	27.393281	-4.940037	
7	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	7	10	10	img0_0-7	10231.794487	1069.314789	10644.289551	2.851640	-1.717653	
8	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	8	10	10	img0_0-8	10437.900894	314.037626	10429.721680	21.930525	-2.908087	
9	..\data\Crystals_Apr_12\Tileset6_subset\Tile_0...	0	9	10	10	img0_0-9	10145.704069	508.349662	10264.887207	17.450511	-3.640575	

In [284]:
```
%matplotlib inline
sb.pairplot(df3, vars=stat_names2)
plt.show()
```



In [286]:
```
%matplotlib notebook
imgutils.plotwithimg(df3, '|img_median|', '|img_std|', imgutils.highlightimgslice, True)
```

Figure 1



In [224]:
```
print(df3['img_mean'].iloc[0])
```

```
8301.389764
```

In [243]:
```
p = imgutils.img_histogram(imgN)
print(p)
```

```
(array([1, 0, 0, ..., 0, 0, 1], dtype=int64), array([ 5808.        ,  5816.48144531,  5824.96289062, ...,
       14476.03710938, 14484.51855469, 14493.        ]))
```

In [244]:
```
s = spstats.gaussian_kde(p[0])
```

In [245]:
```
print(s)
```

```
<scipy.stats.kde.gaussian_kde object at 0x0000000011A6FB70>
```

In [247]:
```
s.pdf()
```

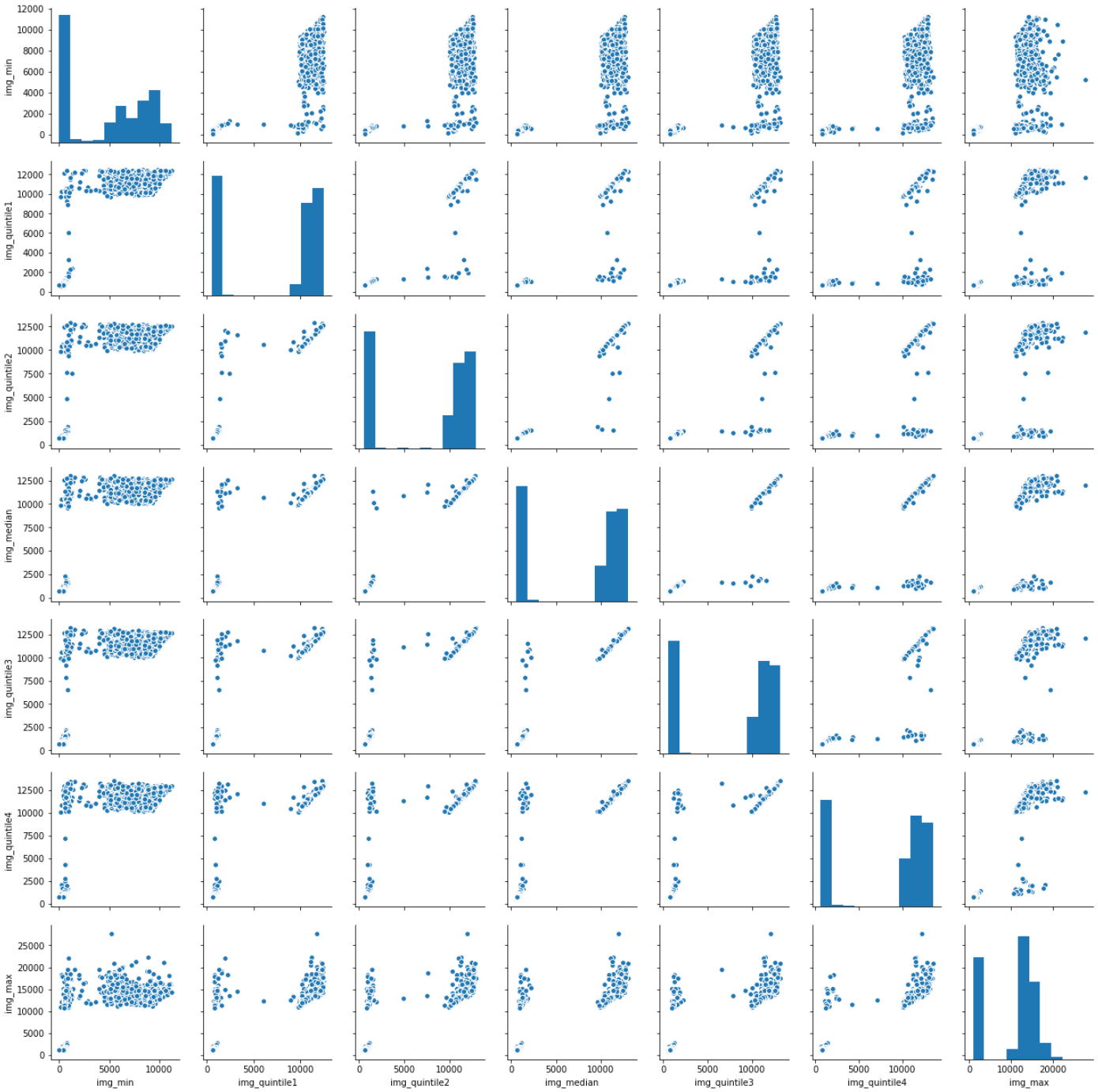
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-247-15cf9801f894> in <module>()
----> 1 s.pdf()

TypeError: pdf() missing 1 required positional argument: 'x'
```

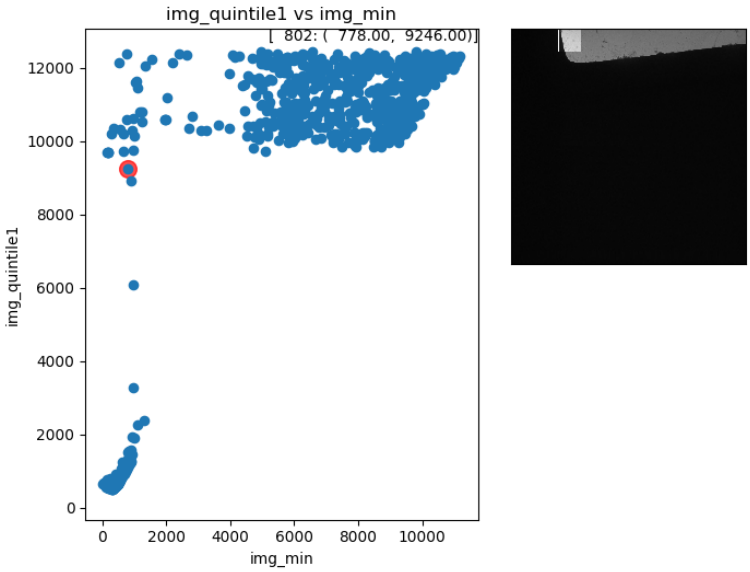
In [258]:
```
stats = imgutils.statfuncs_7numsummary()
df4 = imgutils.slicestats(list(df_imgfiles['filename'][:]), 10, 10, stats)
```

In [264]:
```
stat_names = imgutils.stat_names(stats)
```

In [268]:
```
%matplotlib inline
sb.pairplot(df4, vars=stat_names)
plt.show()
```

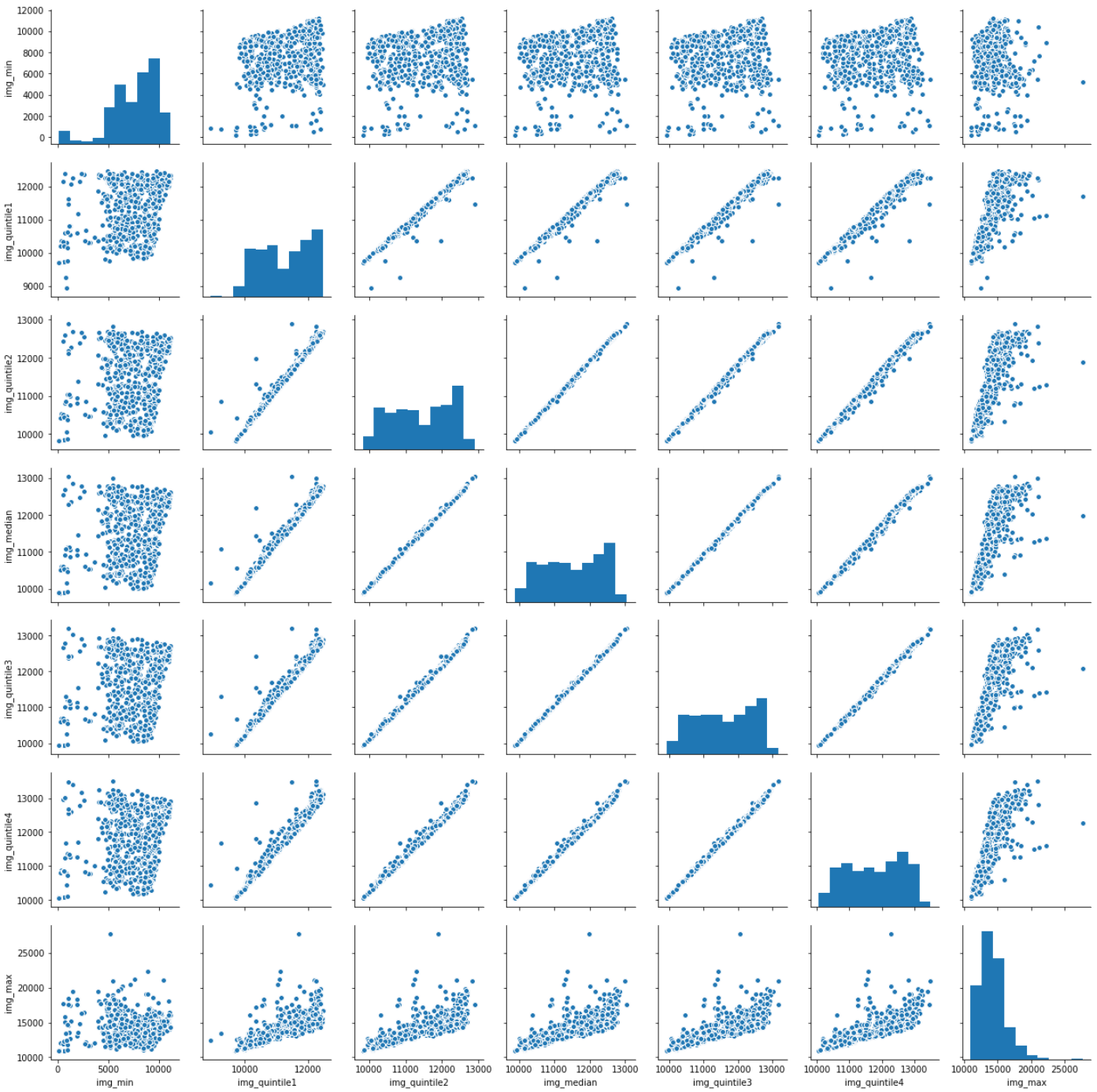


In [273]:
```
%matplotlib notebook
imgutils.plotwithimg(df4, 'img_min', 'img_quintile1', imgutils.highlightimgslice, True)
```



In [274]:
```
df5 = df4[df4['img_quintile1']>8000]
```

In [275]:
```
%matplotlib inline
sb.pairplot(df5, vars=stat_names)
plt.show()
```



Conclusions:

- spend a lot of time on 'feature engineering', but on this set it is really tough
- Need to filter out the black images
- the images have different scaling, so need proper rescaling
- added many new statistics, some of them are useless otthers add some value
- hard to detect multi-modality
- continue to focus on full pipeline
- consider showing histogram in interactive graph and also a zoom in of the image

In [ ]:
```
```