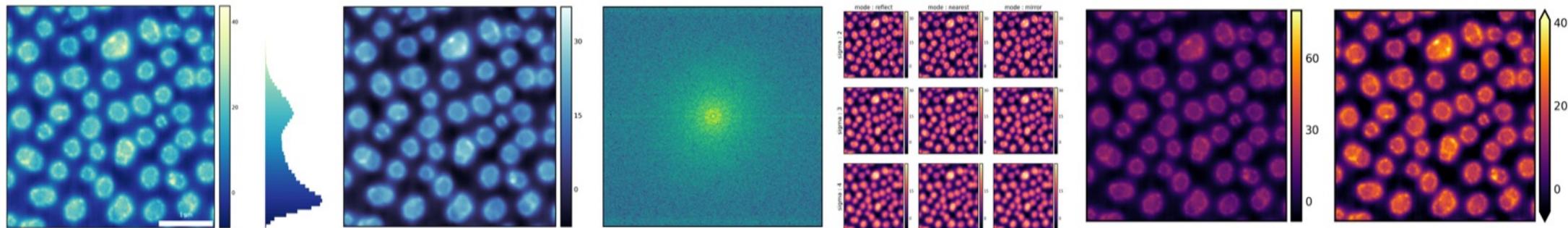


seaborn-image: image data visualization



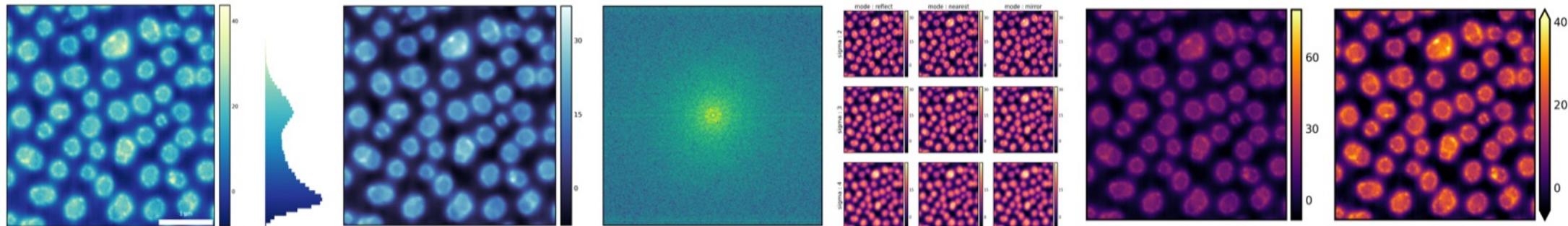
High level interface *based on matplotlib* to draw attractive and informative images quickly and effectively.

Heavily inspired by *seaborn*

GitHub - <https://github.com/SarthakJariwala/seaborn-image>

seaborn-image: image data visualization

 Tests passing  codecov 95%  pypi v0.4.3  docs passing  code style black



- Provide a new domain specific API
- Keeps matplotlib at the core
- Utilizes the well-tested backend of matplotlib
- Image Viz at the core
- Complex image visualizations easy
- Easy to change settings and defaults

seaborn-image Examples

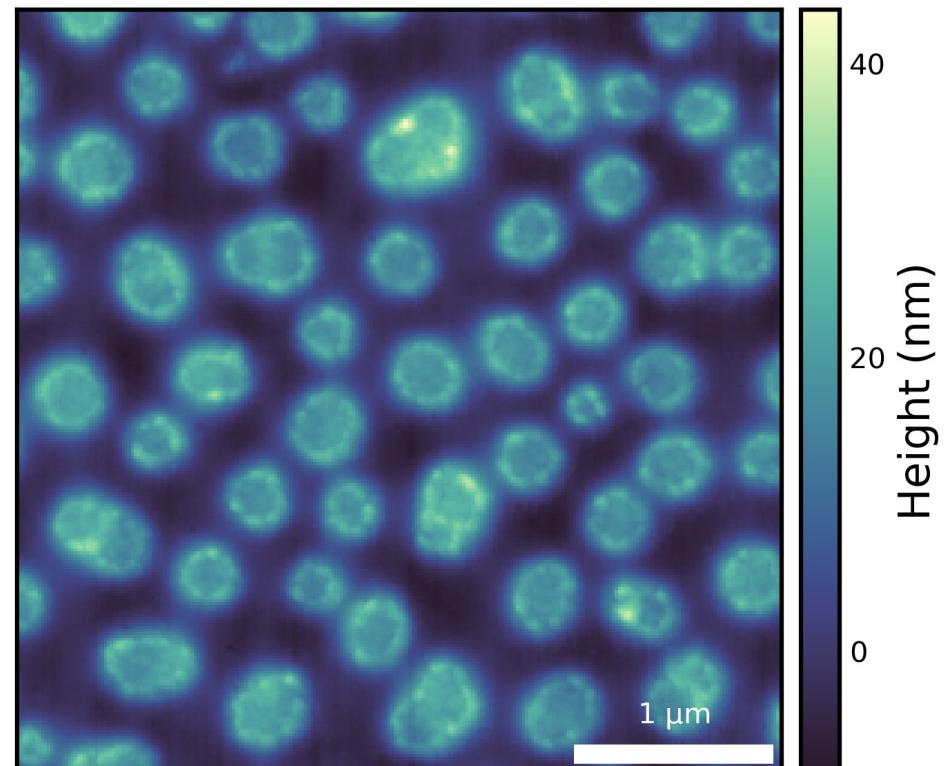
Basic image visualization takes only one line of code.

```
● ● ●

import seaborn_image as sns

# sample dataset
polymer = sns.load_image("polymer")

# plot
ax = sns.imgplot(polymer,
                  dx=15, # physical size/pixel
                  units="nm", # units
                  cbar_label="Height (nm)"
                  )
```



Modify Image and Related Properties



```
import seaborn_image as sns

# Set Scalebar properties
sns.set_scalebar(color='white',
                  location='lower right',
                  width_fraction=0.025,
                  length_fraction=0.3
                 )

# Set Image properties
sns.set_image(cmap='deep',
              origin='lower',
              interpolation='nearest',
              despine=False
             )
```

Alter the defaults easily

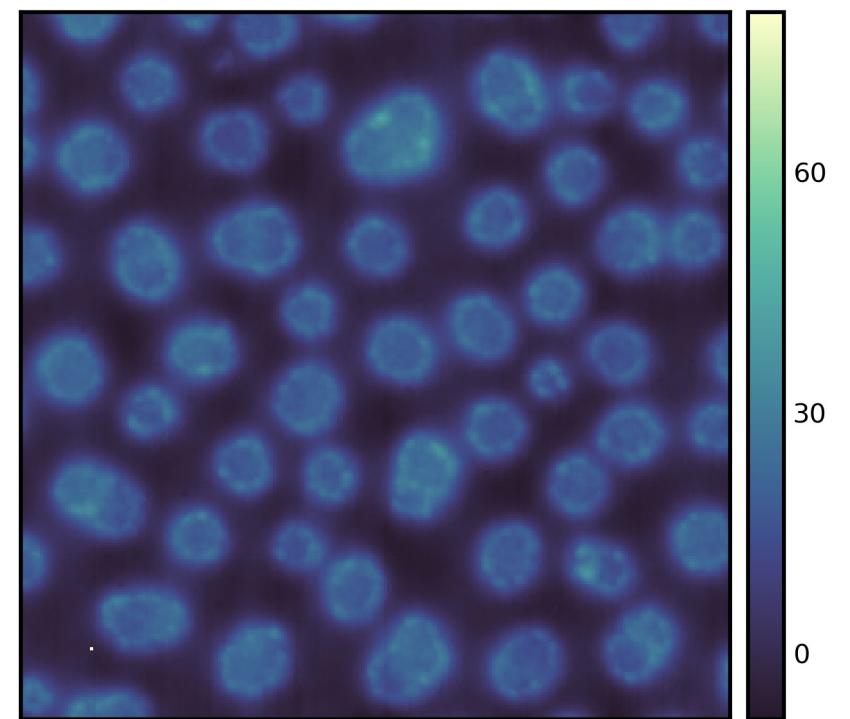
Settings :

set_scalebar()
set_image()
set_context()
set_save_context()
reset_defaults()

Correcting Outliers

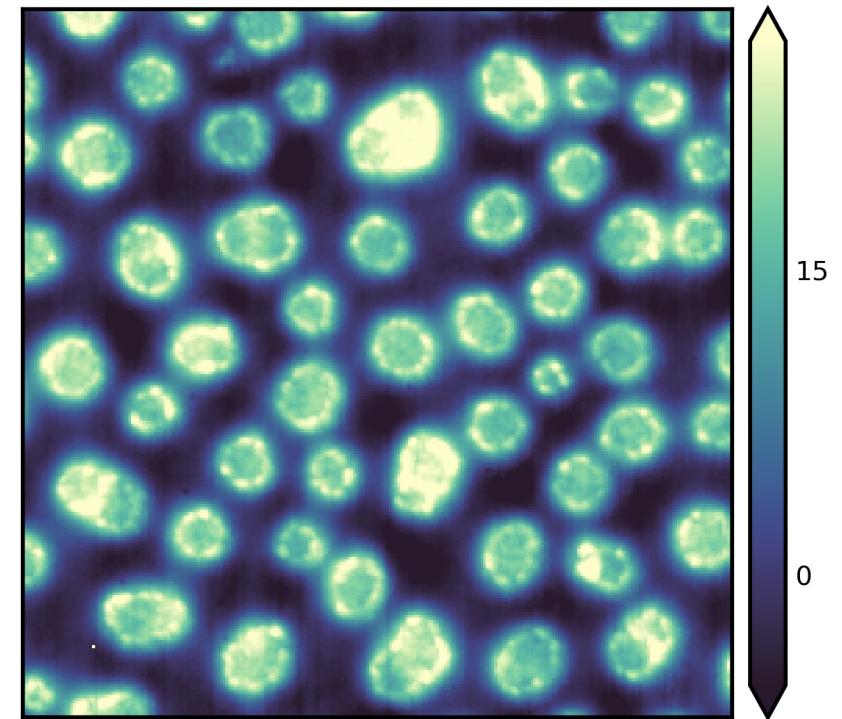
Real world data is never perfect...

```
● ● ●  
import seaborn_image as sns  
  
# sample dataset  
polymer_out = sns.load_image("polymer outliers")  
  
# image with a single outlier data point  
ax = sns.imgplot(polymer_out)
```



Correcting Outliers

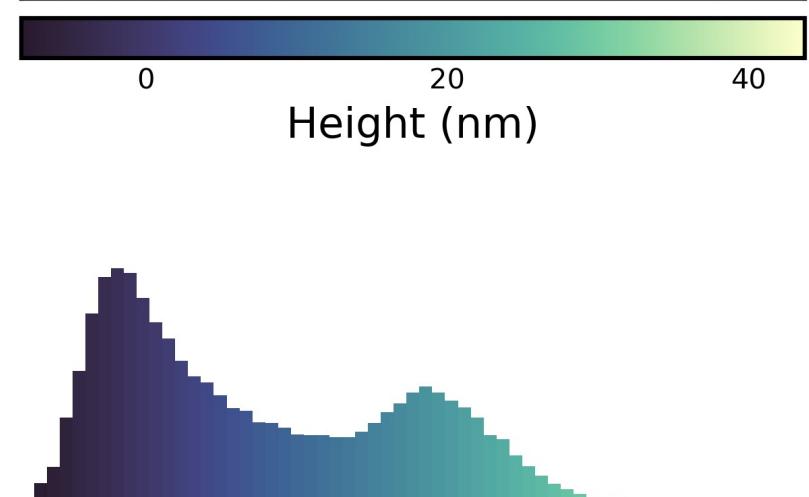
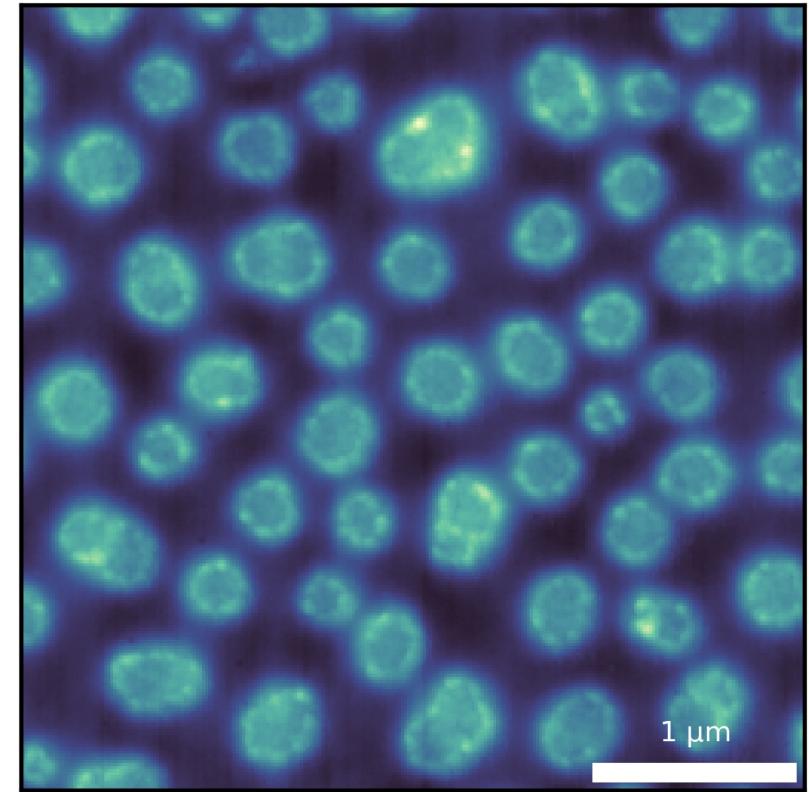
```
● ● ●  
import seaborn_image as sns  
  
# sample dataset  
polymer_out = sns.load_image("polymer outliers")  
  
# image with a single outlier data point  
ax = sns.imgplot(polymer_out, robust=True)
```



Correcting color bar to 0.5 – 99.5 percentile of the data
(can be modified using `perc` parameter)

Visualizing Image Distribution

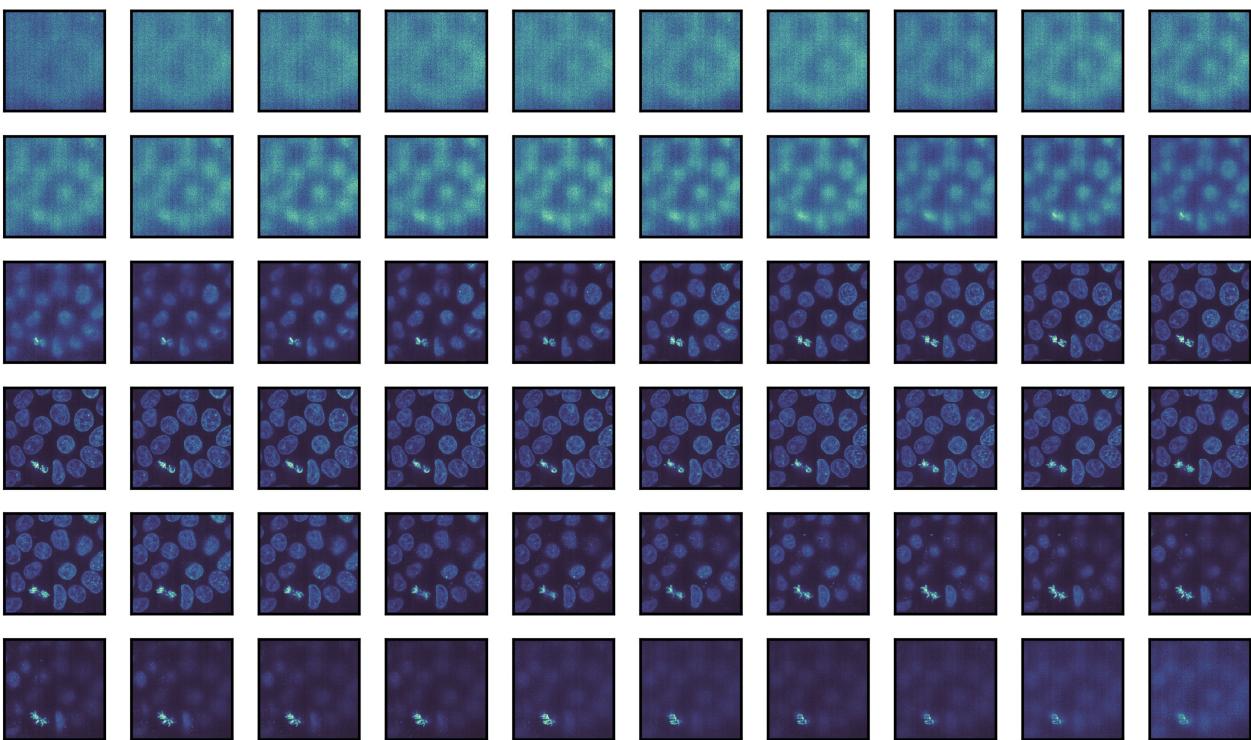
```
● ● ●  
import seaborn_image as sns  
  
# sample dataset  
polymer = sns.load_image("polymer")  
  
# image with a histogram  
fig = sns.imghist(polymer,  
                  orientation="h",  
                  dx=15,  
                  units="nm",  
                  cbar_label="Height (nm)"  
)
```



Multi-Dimensional Data



```
import seaborn_image as sns  
  
# sample 3D dataset  
cells = sns.load_image("cells")  
  
g = sns.ImageGrid(cells, height=1, col_wrap=10, cbar=False)
```



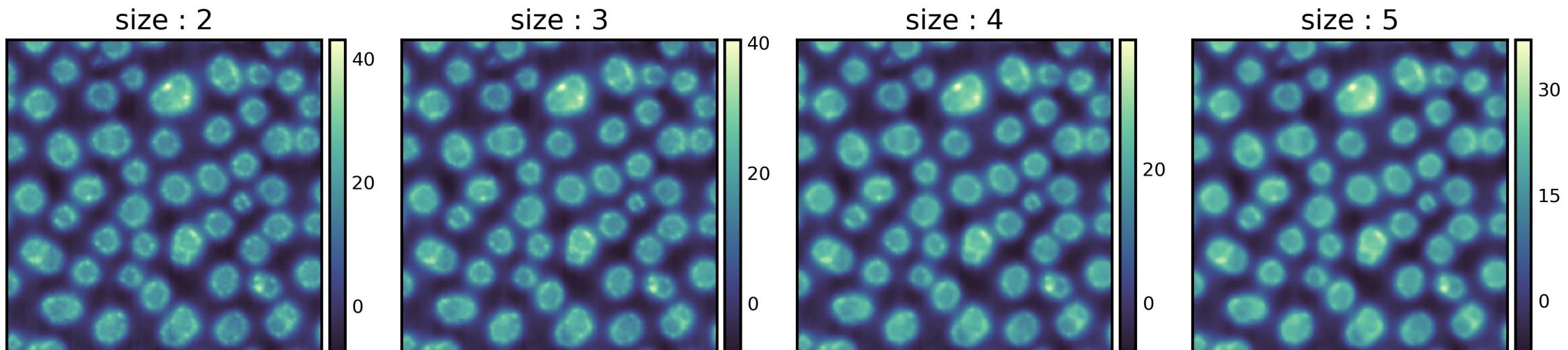
Exploring Relationships b/w Parameters



```
import seaborn_image as sns  
  
polymer = sns.load_image("polymer")  
  
g = sns.FilterGrid(polymer, "median", col="size", size=[2,3,4,5])
```

Apply "median" filter to your image.

Size → 2, 3, 4, 5



`map_func`

Coming soon in v0.5



```
from skimage.exposure import adjust_gamma  
  
import seaborn_image as isns  
  
cells = isns.load_image("cells")  
  
g = isns.ImageGrid(  
    cells,  
    map_func=adjust_gamma,  
    map_func_kwargs={"gamma": 0.5},  
    cbar=False,  
    height=1,  
    col_wrap=10,  
)
```



```
from scipy.ndimage import median_filter, sobel  
from skimage.filters import gaussian  
  
import seaborn_image as isns  
  
# sample datasets  
pl = isns.load_image("fluorescence")  
polymer = isns.load_image("polymer")  
  
# functions to transform the input image  
map_func = [gaussian, median_filter, sobel]  
map_func_kw_args = [{"sigma": 1.5}, {"size": 10}, None]  
  
g = isns.ImageGrid(  
    [pl, polymer],  
    map_func=map_func,  
    map_func_kw_args=map_func_kw_args,  
    col_wrap=2,  
    cmap="inferno",  
)
```

Settings

set_context

set_scalebar

set_image

Plotting

imgplot

ImageGrid

imghist

FilterGrid

fftplot

rgbplot

filterplot

Tutorial

2-D Image Visualization

- [Image Visualization](#)
 - [Add a scalebar](#)
 - [Modify colorbar properties](#)
 - [Modify colormaps](#)
 - [Correct for outliers](#)
 - [Despine image axes](#)

Visualize the Distribution within Image Data

- [Visualize Image Distribution](#)

Working with RGB Images

- [RGB Images](#)
 - [Visualize RGB images](#)
 - [Split and visualize individual channels](#)

Working with Multi-dimensional Images

- [Multi-dimensional images](#)
 - [Plot specific slices from 3-D data](#)
 - [Index along different axis](#)
 - [Variable step sizes and start/end points](#)
 - [Variable column length and figure size](#)
 - [Map a function to transform the 3-D data](#)

Working with a Collection of Images

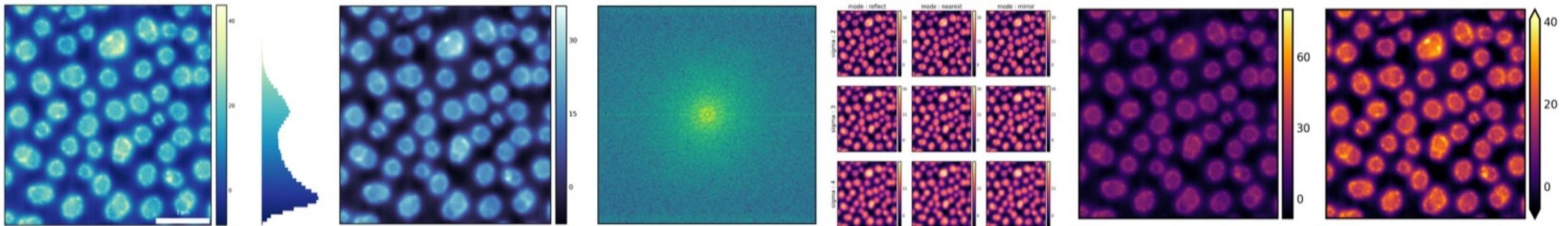
Detailed documentation

- Examples (*online and in documentation strings*)
- Tutorials
- API Reference

<https://seaborn-image.readthedocs.io/en/latest/>

seaborn-image: image data visualization

Tests passing codecov 95% pypi v0.4.3 docs passing code style black



<https://github.com/SarthakJariwala/seaborn-image>

Detailed documentation

- Examples (*online and in documentation strings*)
- Tutorials
- API Reference

Twitter -
@JariwalaSarthak

Thank You!