In [1]:
```
import numpy as np
import pandas as pd
import pathlib
import matplotlib.pyplot as plt
import matplotlib as mpl
import scanpy as sc
import squidpy as sq
```

In [2]:
```
def DissectSegRes(df):
    tmps = []
    for row in df.iterrows():
        if row[1]['segmentation_label'] == 0:
            continue
        for idx,i in enumerate(row[1]['segmentation_centroid']):
            tmps.append(list(i)+[row[0],row[0]+'_{}'.format(idx),row[1]['segmentation_label']])
    return pd.DataFrame(tmps,columns=['x','y','spot_index','cell_index','cell_nums'])

def PlotVisiumCells(adata,annotation_list,size=0.8,alpha_img=0.3,subset=None,ax=None):
    merged_df = adata.uns['cell_locations']
    test = sc.AnnData(np.zeros(merged_df.shape), obs=merged_df)
    test.obsm['spatial'] = merged_df[["x", "y"]].to_numpy()
    test.uns = adata.uns

    if subset is not None:
        test = test[test.obs[annotation_list].isin(subset)]

    sc.pl.spatial(
        test,
        color=annotation_list,
        size=size,
        frameon=False,
        alpha_img=alpha_img,
        show=False,
        ax=ax,title=''
    )

    sf = adata.uns['spatial']['V1_Adult_Mouse_Brain']['scalefactors']['tissue_hires_scalef']
    spot_radius = adata.uns['spatial']['V1_Adult_Mouse_Brain']['scalefactors']['spot_diameter_fullres']/2
    for sloc in adata.obsm['spatial']:
        rect = mpl.patches.Circle(
            (sloc[0] * sf, sloc[1] * sf),
            spot_radius * sf,
            ec="grey",
            lw=1,
            fill=False
        )
        ax.add_patch(rect)
    ax.axes.xaxis.label.set_visible(False)
    ax.axes.yaxis.label.set_visible(False)
```

In [3]:
```
img = sq.datasets.visium_hne_image()
adata = sq.datasets.visium_hne_adata()
```

In [4]:
```
img
```

Out[4]: ImageContainer object with 1 layer:

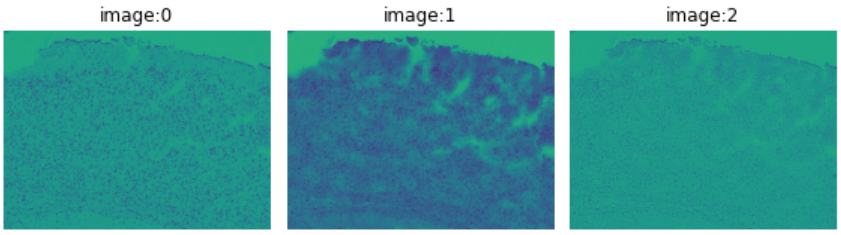
image: *y* (11757), *x* (11291), *z* (1), *channels* (3)

In [5]:
```
inset_x = int(img.shape[0]*(3.7/12.6)) # column
inset_y = int(img.shape[1]*(1.6/13)) # row
inset_sx = int(img.shape[0]*(2.5/12.6))
inset_sy = int(img.shape[1]*(2/13))
```

In [6]:
```
inset_x,inset_y, inset_sx, inset_sy
```

Out[6]: (3452, 1389, 2332, 1737)

In [7]:
```
crop = img.crop_corner(inset_y,inset_x,(inset_sy,inset_sx))
crop.show(channelwise=True,layer='image')
```



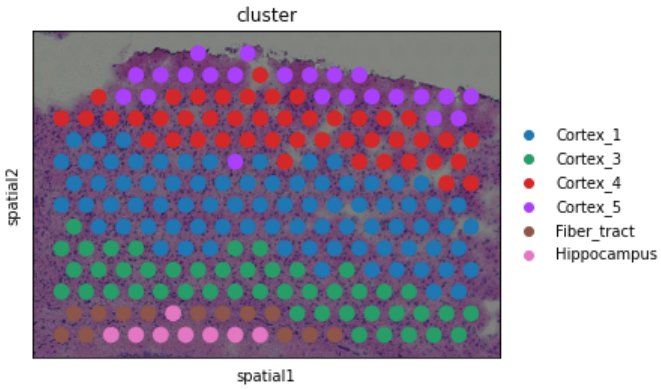
In [8]:
```
crop
```

Out[8]: ImageContainer object with 1 layer:

image: *y* (1737), *x* (2332), *z* (1), *channels* (3)

In [9]:
```
adata_Mop = crop.subset(adata)
```

In [10]:
```
sc.pl.spatial(adata_Mop, color="cluster", size=1)
```



In [11]:
```
from stardist.models import StarDist2D
# Import the recommended normalization technique for stardist.
from csbdeep.utils import normalize
```

In [12]:
```
def stardist_2D_versatile_he(img, nms_thresh=None, prob_thresh=None):
    #axis_norm = (0,1)   # normalize channels independently
    axis_norm = (0,1,2) # normalize channels jointly
    # Make sure to normalize the input image beforehand or supply a normalizer to the prediction function.
    # this is the default normalizer noted in StarDist examples.
    img = normalize(img, 1, 99.8, axis=axis_norm)
    model = StarDist2D.from_pretrained('2D_versatile_he')
    labels, _ = model.predict_instances(img, nms_thresh=nms_thresh, prob_thresh=prob_thresh)
    return labels
StarDist2D.from_pretrained('2D_versatile_he')
```

```
Found model '2D_versatile_he' for 'StarDist2D'.
2022-08-11 15:44:01.417989: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is opt
imized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-c
ritical operations:  SSE4.1 SSE4.2 AVX AVX2 AVX512F FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2022-08-11 15:44:09.750677: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1525] Created device /job:local
host/replica:0/task:0/device:GPU:0 with 5698 MB memory:  -> device: 0, name: Tesla V100-SXM2-16GB, pci bus id:
0000:61:00.0, compute capability: 7.0
2022-08-11 15:44:09.790631: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1525] Created device /job:local
host/replica:0/task:0/device:GPU:1 with 1401 MB memory:  -> device: 1, name: Tesla V100-SXM2-16GB, pci bus id:
0000:62:00.0, compute capability: 7.0
2022-08-11 15:44:09.798215: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1525] Created device /job:local
host/replica:0/task:0/device:GPU:2 with 1403 MB memory:  -> device: 2, name: Tesla V100-SXM2-16GB, pci bus id:
0000:89:00.0, compute capability: 7.0
2022-08-11 15:44:09.828347: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1525] Created device /job:local
host/replica:0/task:0/device:GPU:3 with 1403 MB memory:  -> device: 3, name: Tesla V100-SXM2-16GB, pci bus id:
0000:8a:00.0, compute capability: 7.0
Loading network weights from 'weights_best.h5'.
Loading thresholds from 'thresholds.json'.
Using default values: prob_thresh=0.692478, nms_thresh=0.3.
```

Out[12]:
```
StarDist2D(2D_versatile_he): YXC → YXC
 ├── Directory: None
```

```
└── Config2D(n_dim=2, axes='YXC', n_channel_in=3, n_channel_out=33, train_checkpoint='weights_best.h5', train_checkpoint_last='weights_last.h5', train_checkpoint_epoch='weights_now.h5', n_rays=32, grid=(2, 2), backbone='unet', n_classes=None, unet_n_depth=3, unet_kernel_size=[3, 3], unet_n_filter_base=32, unet_n_conv_per_depth=2, unet_pool=[2, 2], unet_activation='relu', unet_last_activation='relu', unet_batch_norm=False, unet_dropout=0.0, unet_prefix='', net_conv_after_unet=128, net_input_shape=[None, None, 3], net_mask_shape=[None, None, 1], train_shape_completion=False, train_completion_crop=32, train_patch_size=[512, 512], train_background_reg=0.0001, train_foreground_only=0.9, train_sample_cache=True, train_dist_loss='mae', train_loss_weights=[1, 0.1], train_class_weights=(1, 1), train_epochs=200, train_steps_per_epoch=200, train_learning_rate=0.0003, train_batch_size=8, train_n_val_patches=3, train_tensorboard=True, train_reduce_lr={'factor': 0.5, 'patience': 50, 'min_delta': 0}, use_gpu=False)
```

In [13]:

```
sq.im.segment(
    img=crop,
    layer="image",
    channel=None,
    method=stardist_2D_versatile_he,
    layer_added='segmented_stardist_default',
    prob_thresh=0.5
)
```

```
Found model '2D_versatile_he' for 'StarDist2D'.
Loading network weights from 'weights_best.h5'.
Loading thresholds from 'thresholds.json'.
Using default values: prob_thresh=0.692478, nms_thresh=0.3.
2022-08-11 15:44:13.502405: I tensorflow/stream_executor/cuda/cuda_dnn.cc:368] Loaded cuDNN version 8302
```

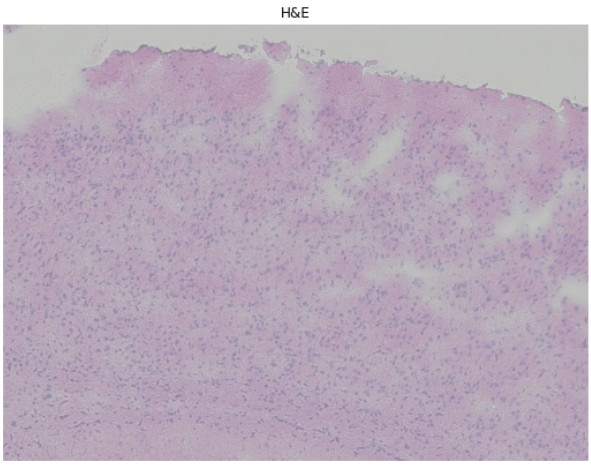
In [14]:

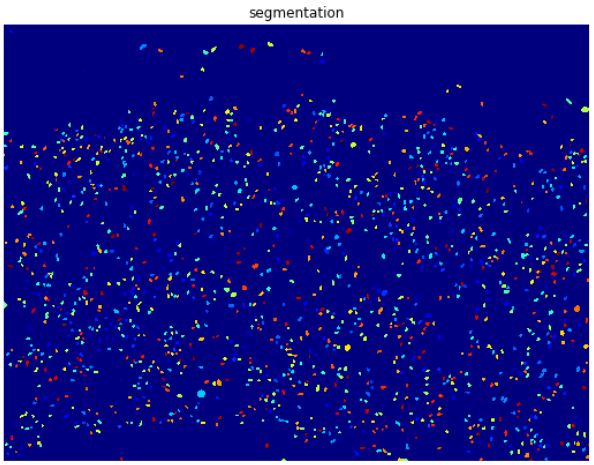
```
print(crop)
print(f"Number of segments in crop: {len(np.unique(crop['segmented_stardist_default']))}")
```

```
ImageContainer[shape=(1737, 2332), layers=['image', 'segmented_stardist_default']]
Number of segments in crop: 1563
```

In [15]:

```
fig, axes = plt.subplots(1, 2,figsize=(20,20))
crop.show("image", ax=axes[0],alpha=0.5)
_ = axes[0].set_title("H&E")
crop.show("segmented_stardist_default", cmap="jet", interpolation="none", ax=axes[1])
_ = axes[1].set_title("segmentation")
```





Using relative coordinates (the default setting in squidpy)

squidpy/im/_feature_mixin.py:

```
    # relative coordinates

    y = (y - np.min(y)) / (np.max(y) - np.min(y))
    x = (x - np.min(x)) / (np.max(x) - np.min(x))
    # coordinates in the uncropped image

    y = coord.slice[0].start + (y_slc.stop - y_slc.start) * y
    x = coord.slice[1].start + (x_slc.stop - x_slc.start) * x
```

In [16]:

```
features_kwargs = {
    "segmentation": {
        "label_layer": "segmented_stardist_default",
        "props": ["label", "centroid"]
```

```
    }
}
# calculate segmentation features
sq.im.calculate_image_features(
    adata_Mop,
    crop,
    layer="image",
    key_added="image_features",
    features_kwargs=features_kwargs,
    features="segmentation",
    mask_circle=True
)
```

```
auto.py (22): IProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.i
o/en/stable/user_install.html
  0%|
| 0/215 [00:00<?, ?/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
  5%|
| 11/215 [00:02<00:36,  5.51/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 12%|
| 25/215 [00:04<00:33,  5.64/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 18%|
| 38/215 [00:06<00:31,  5.68/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 20%|
| 44/215 [00:08<00:32,  5.30/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 22%|
| 48/215 [00:08<00:31,  5.23/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 27%|
| 59/215 [00:10<00:27,  5.64/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 35%|
| 76/215 [00:13<00:24,  5.76/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 41%|
| 89/215 [00:16<00:22,  5.67/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 44%|
| 95/215 [00:17<00:21,  5.63/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 47%|
| 101/215 [00:18<00:20,  5.68/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 47%|
| 102/215 [00:18<00:19,  5.67/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 54%|
| 116/215 [00:21<00:17,  5.76/s]_featu
re_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 55%|
| 118/215 [00:21<00:17,  5.64/s]_feat
ure_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 60%|
| 128/215 [00:23<00:15,  5.74/s]_
feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 70%|
| 151/215 [00:27<00:11,
5.67/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 71%|
| 152/215 [00:27<00:11,
5.66/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 80%|
| 172/215 [00:30
<00:07,  5.73/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 81%|
| 175/215 [00:
31<00:07,  5.64/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 84%|
```

```
| 181/215 [0
0:32<00:06,  5.54/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 89%|
| 192/21
5 [00:34<00:04,  5.65/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 90%|
| 193/2
15 [00:34<00:03,  5.58/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 90%|
| 194/
215 [00:34<00:03,  5.59/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 91%|
| 196/
215 [00:35<00:03,  5.69/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
 92%|
| 197/
215 [00:35<00:03,  5.70/s]_feature_mixin.py (346): invalid value encountered in true_divide
_feature_mixin.py (347): invalid value encountered in true_divide
100%|
| 215/215 [00:38<00:00,  5.57/s]
```

In [18]:
```
adata_Mop.obsm["image_features"].head()
```

Out[18]:

	segmentation_label	segmentation_centroid
AAACAATCTACTAGCA-1	1	[[nan, nan]]
AAACGAAGAACATACC-1	5	[[5579.824012330447, 1926.0], [5592.9611046823...
AAAGACCCAAGTCGCG-1	2	[[4470.0, 2494.0], [4559.0, 2405.0]]
AAAGTCACTGATGTAA-1	2	[[4834.0, 2494.0], [4745.0, 2405.0]]
AACACGACTGTACTGA-1	7	[[5728.0, 2371.1059355145994], [5639.0, 2337.8...

As shown in first row, because of the normalization of "np.max(y) - np.min(y)" in denominator, relative coordinate will return nan if there is only one cell in spot

In [ ]:
```
```

In [19]:
```
adata_Mop.uns['cell_locations'] = DissectSegRes(adata_Mop.obsm['image_features'])
```

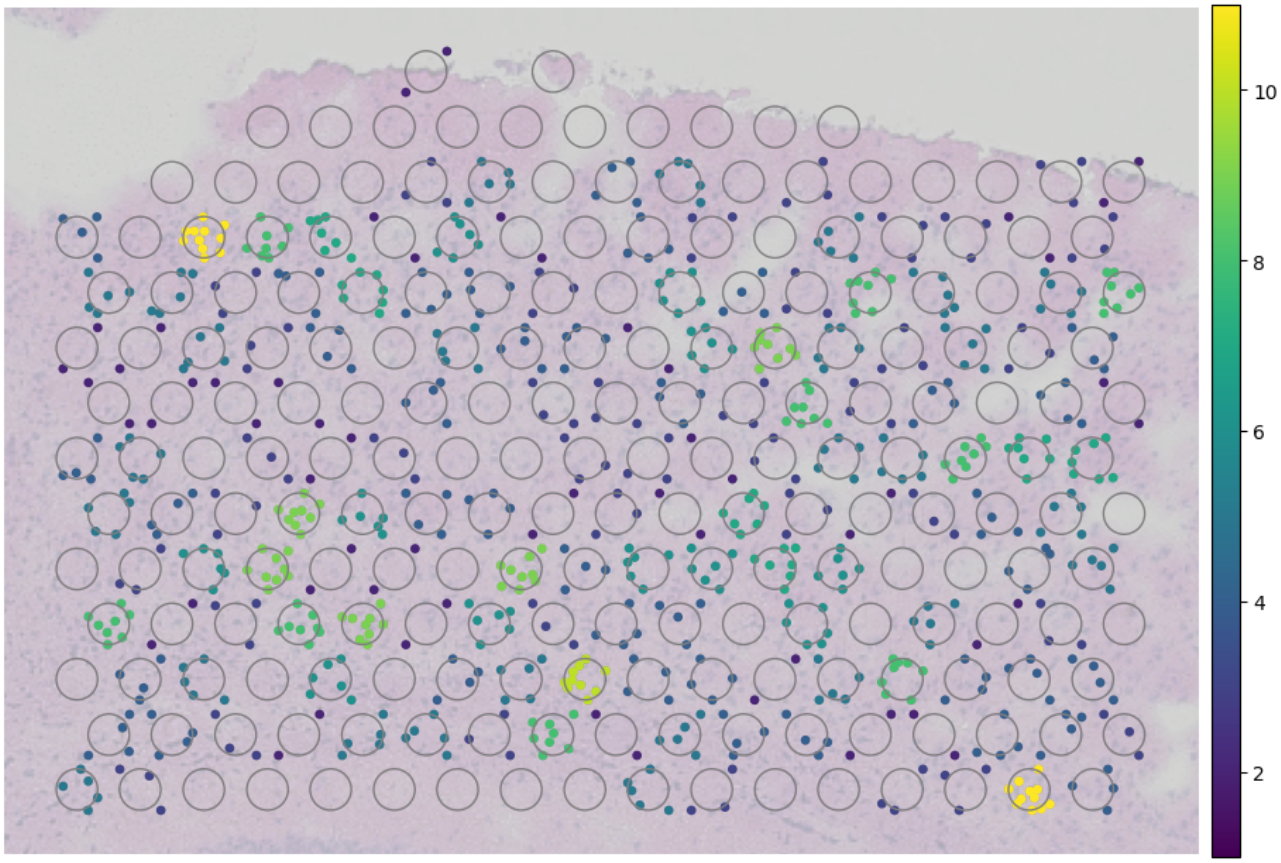
In [20]:
```
adata_Mop.uns['cell_locations'].head()
```

Out[20]:

	x	y	spot_index	cell_index	cell_nums
0	NaN	NaN	AAACAATCTACTAGCA-1	AAACAATCTACTAGCA-1_0	1
1	5579.824012	1926.000000	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_0	5
2	5592.961105	1951.672096	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_1	5
3	5660.000000	1967.866139	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_2	5
4	5637.048979	2015.000000	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_3	5

In [21]:
```
fig, ax = plt.subplots(1,1,figsize=(12, 8),dpi=100)
PlotVisiumCells(adata_Mop,"cell_nums",0.23,0.35,ax=ax)
```

```
1501143866.py (12): X.dtype being converted to np.float32 from float64. In the next version of anndata (0.9) c
onversion will not be automatic. Pass dtype explicitly to avoid this warning. Pass `AnnData(X, dtype=X.dtype,
...)` to get the future behavour.
anndata.py (121): Transforming to str index.
```



As shown in the above figure, relative coordinates look weird when visualizing the segmentation results with spatial image as background

In [ ]:

Using exact nuclei coordinates as in squidpy/im/_feature_mixin.py of pull request

squidpy/im/_feature_mixin.py:

```
# exact coordinates
y = coord.slice[0].start + y
x = coord.slice[1].start + x
```

In [15]:

```
features_kwargs = {
    "segmentation": {
        "label_layer": "segmented_stardist_default",
        "props": ["label", "centroid"]
    }
}
# calculate segmentation features
sq.im.calculate_image_features(
    adata_Mop,
    crop,
    layer="image",
    key_added="image_features",
    features_kwargs=features_kwargs,
    features="segmentation",
    mask_circle=True
)
```

```
auto.py (22): IProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html
100%|██████████| 215/215 [00:59<00:00,  3.63/s]
```

In [ ]:

In [16]:
```
adata_Mop.obsm["image_features"].head()
```

Out[16]:

	segmentation_label	segmentation_centroid
AAACAATCTACTAGCA-1	1	[[4155.295302013423, 1601.9261744966443]]
AAACGAAGAACATACC-1	5	[[5616.569948186529, 1946.8134715025906], [562...
AAAGACCCAAGTCGCG-1	2	[[4506.21875, 2441.5625], [4523.902857142857, ...
AAAGTCACTGATGTAA-1	2	[[4771.435294117647, 2485.635294117647], [4759...
AACACGACTGTACTGA-1	7	[[5715.401273885351, 2342.6751592356686], [564...

In [17]:
```
adata_Mop.uns['cell_locations'] = DissectSegRes(adata_Mop.obsm['image_features'])
```

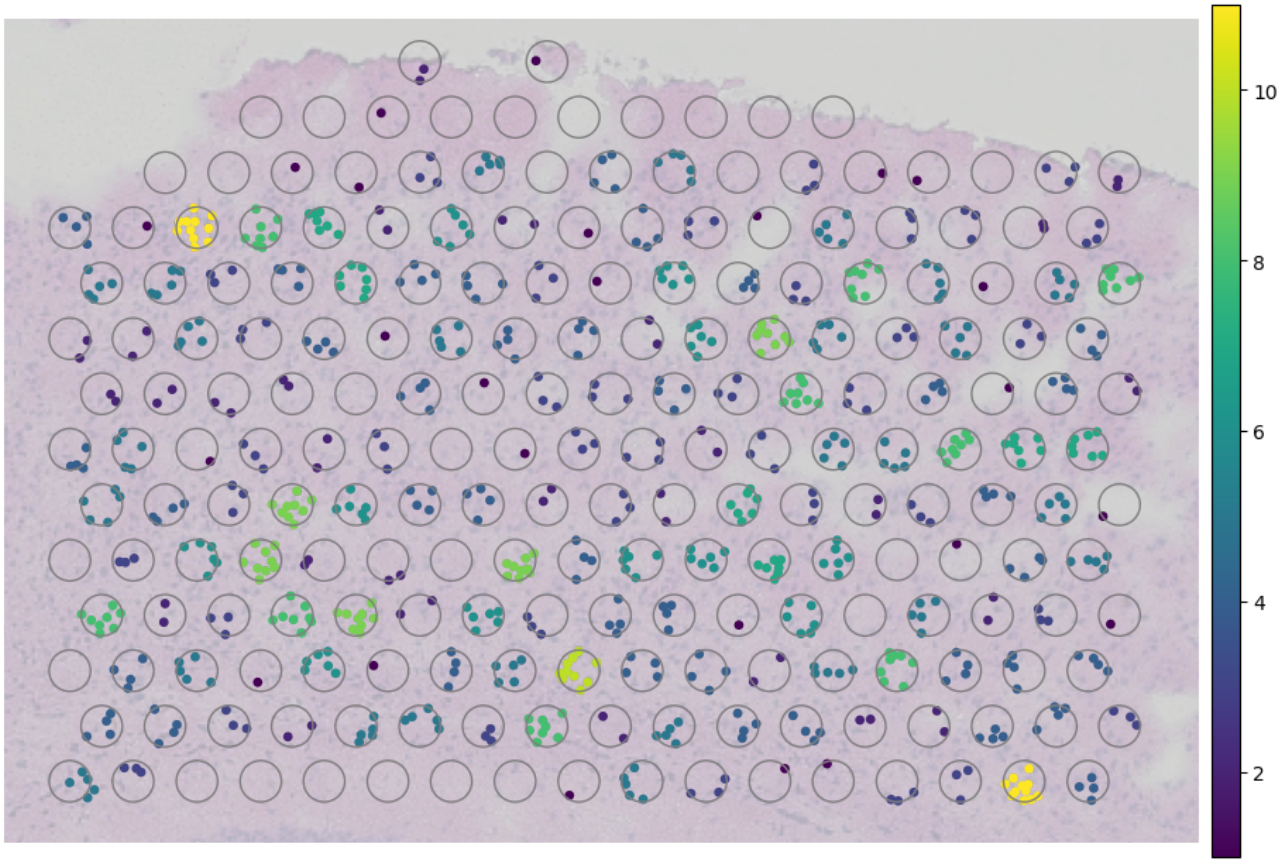
In [19]:
```
adata_Mop.uns['cell_locations'].head()
```

Out[19]:

	x	y	spot_index	cell_index	cell_nums
0	4155.295302	1601.926174	AAACAATCTACTAGCA-1	AAACAATCTACTAGCA-1_0	1
1	5616.569948	1946.813472	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_0	5
2	5622.854962	1962.595420	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_1	5
3	5654.927536	1972.550725	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_2	5
4	5643.947368	2001.526316	AAACGAAGAACATACC-1	AAACGAAGAACATACC-1_3	5

In [21]:
```
fig, ax = plt.subplots(1,1,figsize=(12, 8),dpi=100)
PlotVisiumCells(adata_Mop,"cell_nums",0.23,0.35,ax=ax)
```

```
1351889192.py (3): X.dtype being converted to np.float32 from float64. In the next version of anndata (0.9) co
nversion will not be automatic. Pass dtype explicitly to avoid this warning. Pass `AnnData(X, dtype=X.dtype,
...)` to get the future behavour.
anndata.py (121): Transforming to str index.
```



In [ ]:
```
```

In [ ]:

In [ ]: