Mapping scRNAseq to 3D coordinate space

The aim is to spatially map single cell RNA-seq (scRNAseq) to a 3D spatial plane such that the cells are spatially arranged analogously to how they would be structured in-vivo.

This is done without prior knowledge of cell locations for the scRNAseq cells.

The training set would include all information from a Spatial Transcriptomic (ST) measurement, including gene expression per spatial coordinate overlayed on histology pixels.

ST data from the Allen Brain Institute is used from:

https://github.com/almaan/spacetx/

This is purposed for the Competition from the Allen Brain Institute:

https://labshare.cshl.edu/shares/gillislab/resource/CellTypeMapping_2022/

Able to run on 16 vCPUs, 128 GB RAM with PyTorch 1.11 (with Intel® MKL-DNN/MKL)

In [1]:
```
# !wget https://labshare.cshl.edu/shares/gillislab/resource/CellTypeMapping_
# !wget https://github.com/almaan/spacetx/raw/master/data/Allen-1.h5ad.gz
# !wget https://github.com/almaan/spacetx/raw/master/data/Allen-2.h5ad.gz
# !wget https://github.com/almaan/spacetx/raw/master/data/Allen-3.h5ad.gz
# !wget https://github.com/almaan/spacetx/raw/master/data/Allen-4.h5ad.gz

# !gunzip Allen-1.h5ad.gz
# !gunzip Allen-2.h5ad.gz
# !gunzip Allen-3.h5ad.gz
# !gunzip Allen-4.h5ad.gz
```

In [2]:
```
import h5py
import torch
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import math
import torch.nn as nn
import gc
import scanpy as sc
from scipy import ndimage
import random
import torch.nn.functional as F
import torchvision.transforms.functional as TF
import leidenalg
import scanorama
import anndata as AnnData
import scvi
```

```
Global seed set to 0
```

In [3]:
```
device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')

seed_num = 1
np.random.seed(seed_num)
```

```
random.seed(seed_num)
torch.manual_seed(seed_num)
```

Out[3]:
```
<torch._C.Generator at 0x7ff652965f70>
```

In [4]:
```
window_size = 2
main_axis_size = 800

# Generate a uniform grid of M cells
def to_grid(M):
    grid = np.random.uniform(size=(M.shape[1],2))
    return grid

# Put gene expression as new "channels" on top of uniform grid
def to_kde_grid(M,grid,main_axis_size=main_axis_size,window_size=window_size
    main_grid = np.zeros(((main_axis_size+4*window_size),(main_axis_size+4*w
    for i in range(M.shape[0]):
        x = np.round(main_axis_size*grid[:,0],0).astype(np.int)+2*window_siz
        y = np.round(main_axis_size*grid[:,1],0).astype(np.int)+2*window_siz
        main_grid[x,y,i] = main_grid[x,y,i]+M[i,:].reshape([M.shape[1],])
        main_grid[:,:,i] = ndimage.gaussian_filter(main_grid[:,:,i], window_
    return main_grid

# Get KNN gene expression "convolutions" from gene expression "channels"
def to_samples(kde_grid,grid,w_size=window_size,type_data="numpy",modality="
    W,H,C = kde_grid.shape
    if modality=="ST":
        x = grid[:,0].astype(np.int)
        y = grid[:,1].astype(np.int)
    else:
        x = np.round(main_axis_size*grid[:,0],0).astype(np.int)+2*window_siz
        y = np.round(main_axis_size*grid[:,1],0).astype(np.int)+2*window_siz

    main_kde_conv = np.zeros((len(x),w_size*2*w_size*2*C))
    for i in range(len(x)):
        main_kde_conv[i,:] = (kde_grid[(x[i]-w_size):(x[i]+w_size),(y[i]-w_s

    if type_data == "tensor":
        main_kde_conv = torch.from_numpy(main_kde_conv).float().to(device)

    return main_kde_conv
```

In [5]:
```
# Read in Allen Brain data
allen_1 = sc.read_h5ad("Allen-1.h5ad")
allen_2 = sc.read_h5ad("Allen-2.h5ad")
allen_3 = sc.read_h5ad("Allen-3.h5ad")
allen_4 = sc.read_h5ad("Allen-4.h5ad")



# Read in Competition data
import h5py
from scipy import sparse
with h5py.File('barseq2.hdf5') as f:
    dim = f['dim'][()]
    genes_old = [g.decode('utf-8') for g in f['dimnames/genes'][()]]
    cells = [c.decode('utf-8') for c in f['dimnames/cells'][()]]
    i = f['matrix/i'][()] - 1
    j = f['matrix/j'][()] - 1
    x = f['matrix/x'][()]
    M = sparse.csc_matrix((x, (i,j)) , shape =dim )

M = M.todense()
```

```
# Select similar genes shared between datasets
genes = [i for i in genes_old if i not in ['Tafa1', 'Tafa2', 'Ccn2']]
allen_1_select = allen_1[:,genes]
allen_2_select = allen_2[:,genes]
allen_3_select = allen_3[:,genes]
allen_4_select = allen_4[:,genes]

M = M[np.where(np.asarray([i in genes for i in genes_old]))[0],:]
```

In [7]:

```
coord_list = [
    np.vstack((allen_1_select.obs["x"]*allen_1_select.uns["tissue_hires_scal
    np.vstack((allen_2_select.obs["x"]*allen_2_select.uns["tissue_hires_scal
    np.vstack((allen_3_select.obs["x"]*allen_3_select.uns["tissue_hires_scal
    np.vstack((allen_4_select.obs["x"]*allen_4_select.uns["tissue_hires_scal
]


pixel_list = [
    allen_1_select.uns["image_hires"],
    allen_2_select.uns["image_hires"],
    allen_3_select.uns["image_hires"],
    allen_4_select.uns["image_hires"]
]


main_pixel_list = [
    allen_1_select.uns["image_hires"],
    allen_2_select.uns["image_hires"],
    allen_3_select.uns["image_hires"],
    allen_4_select.uns["image_hires"]
]


gex_list = [
    allen_1_select.X,
    allen_2_select.X,
    allen_3_select.X,
    allen_4_select.X
]


for i in range(len(gex_list)):
    gex_list[i] = torch.from_numpy(gex_list[i]).float().to(device)
    pixel_list[i] = to_samples(pixel_list[i],coord_list[i],16,"numpy","ST")
    pixel_list[i] = torch.from_numpy(pixel_list[i]).float().to(device)
    print(pixel_list[i].shape)
    print(gex_list[i].shape)
```

```
torch.Size([2669, 3072])
torch.Size([2669, 106])
torch.Size([3571, 3072])
torch.Size([3571, 106])
torch.Size([2741, 3072])
torch.Size([2741, 106])
torch.Size([3354, 3072])
torch.Size([3354, 106])
```

In [ ]:

In [ ]:

In [8]:
```
# Prepare Allen brain dataset



allen_1_grid = np.vstack((allen_1_select.obs["x"]*allen_1_select.uns["tissue
allen_1_select.obs["y"]*allen_1_select.uns["tissue_hires_scalef"]/2000)).T
allen_1_M = allen_1_select.X.T
print(allen_1_grid.shape)
print(allen_1_M.shape)

allen_1_kde_grid = to_kde_grid(allen_1_M,allen_1_grid,main_axis_size)
allen_1_kde_samples = to_samples(allen_1_kde_grid,allen_1_grid,window_size,"

print(allen_1_kde_samples.shape)




allen_2_grid = np.vstack((allen_2_select.obs["x"]*allen_2_select.uns["tissue
allen_2_select.obs["y"]*allen_2_select.uns["tissue_hires_scalef"]/2000)).T
allen_2_M = allen_2_select.X.T
print(allen_2_grid.shape)
print(allen_2_M.shape)

allen_2_kde_grid = to_kde_grid(allen_2_M,allen_2_grid,main_axis_size)
allen_2_kde_samples = to_samples(allen_2_kde_grid,allen_2_grid,window_size,"

print(allen_2_kde_samples.shape)




allen_3_grid = np.vstack((allen_3_select.obs["x"]*allen_3_select.uns["tissue
allen_3_select.obs["y"]*allen_3_select.uns["tissue_hires_scalef"]/2000)).T
allen_3_M = allen_3_select.X.T
print(allen_3_grid.shape)
print(allen_3_M.shape)

allen_3_kde_grid = to_kde_grid(allen_3_M,allen_3_grid,main_axis_size)
allen_3_kde_samples = to_samples(allen_3_kde_grid,allen_3_grid,window_size,"

print(allen_3_kde_samples.shape)



allen_4_grid = np.vstack((allen_4_select.obs["x"]*allen_4_select.uns["tissue
allen_4_select.obs["y"]*allen_4_select.uns["tissue_hires_scalef"]/2000)).T
allen_4_M = allen_4_select.X.T
print(allen_4_grid.shape)
print(allen_4_M.shape)

allen_4_kde_grid = to_kde_grid(allen_4_M,allen_4_grid,main_axis_size)
allen_4_kde_samples = to_samples(allen_4_kde_grid,allen_4_grid,window_size,"

print(allen_4_kde_samples.shape)
```

```
(2669, 2)
(106, 2669)
(2669, 1696)
(3571, 2)
(106, 3571)
(3571, 1696)
(2741, 2)
(106, 2741)
(2741, 1696)
(3354, 2)
(106, 3354)
(3354, 1696)
```

In [9]:
```
# Generate uniform grid
grid = to_grid(M)

# Generate gene expression channels
kde_grid = to_kde_grid(M,grid,main_axis_size)

# Generate KNN convolutions from gene expression channels
kde_samples = to_samples(kde_grid,grid,window_size,"numpy","scRNAseq")
```

In [10]:
```
class VisualFormNet(nn.Module):

    def __init__(self):
        super(VisualFormNet, self).__init__()

        # beta is to learn gene expression
        # spatial signals from scRNAseq only
        self.summary_beta = nn.Linear(10,window_size*2*window_size*2*106,bia

        # gex and pix are to learn transformation parameters that
        # assign based on image similarity to real histology
        self.summary_gex = nn.Linear(len(genes),100,bias=False)
        self.summary_pix = nn.Linear(32*32*3,100,bias=False)

        # output layer
        self.output_gex = nn.Linear(10,3)

        self.eye_beta = torch.eye(10,10).to(device)
        self.eye_pix = torch.eye(32*32*3).to(device)
        self.eye_gex = torch.eye(len(genes)).to(device)

        self.dropout_gex = nn.Dropout(0.6)


    def forward(self, x_list, gex_list, pix_list):




        # Use spatial gene expression to align single cells properly
        spatial_gex_list = []



        # Run
        for i in range(len(gex_list)):
            gex = gex_list[i]
            pix = pix_list[i]
            N,P = gex.shape
            s_pix_2_code = self.summary_pix(self.eye_pix)
            s_gene_2_code = self.dropout_gex(self.summary_gex(self.eye_gex))
            s_pix_2_gex = s_pix_2_code@s_gene_2_code.T
```

```
            gex_pix1,gex_pix2 = pix@s_pix_2_gex,pix@s_pix_2_gex
            spatial_gex_list += [[gex_pix1,gex_pix2,s_pix_2_gex]]




        # Use spatial histology pixels to align single cells properly
        ST_grid = []



        # Find a set of optimal signals that push cells around like Waddingt
        # via linear form model - see https://github.com/AskExplain/Interpre
        x_grid = []


        # Run
        for i in range(len(x_list)):
            x = x_list[i]

            ST_x = x.reshape([x.shape[0],window_size*2,window_size*2,106]).p
            ST_x = torch.mean(ST_x,(2,3)).reshape(ST_x.shape[:2])
            ST_grid += [[ST_x@s_pix_2_gex.T]]

            N,P = x.shape
            beta = self.summary_beta(self.eye_beta).T
            x1,x2 = self.output_gex(x@beta), self.output_gex(x@beta@(beta.T@
            x1,x2 = torch.subtract(x1,torch.mean(x1).to(device)),torch.subtr
            x1,x2 = torch.div(x1,torch.std(x1).to(device)),torch.div(x2,torc
            x1,x2 = torch.subtract(x1,torch.min(x1).to(device)),torch.subtra
            x1,x2 = torch.div(x1,torch.max(x1).to(device)),torch.div(x2,torc
            x_grid += [[x1,x2]]


        return x_grid, ST_grid, spatial_gex_list
```

In [11]:
```
import torch.optim as optim

net = VisualFormNet()
net.to(device)

mseloss = nn.MSELoss()
optimizer = optim.Adam(net.parameters(), lr=1e-2, weight_decay = 0.5)
```

In [12]:
```
# GRID_list is a normalised list of coordinates from 0 to 1
GRID_list = [
    allen_1_grid,
    allen_2_grid,
    allen_3_grid,
    allen_4_grid
]

# Gene expression
M_list = [
          M,
          allen_1_M,
          allen_2_M,
          allen_3_M,
          allen_4_M
         ]
```

```
# Spatial gene expression convolutions from gene "channels"
x_list = [kde_samples,
          allen_1_kde_samples,
          allen_2_kde_samples,
          allen_3_kde_samples,
          allen_4_kde_samples
]



del allen_1_M,allen_2_M,allen_3_M,allen_4_M
gc.collect()

for i in range(len(x_list)):
    x_list[i] = torch.from_numpy(x_list[i]).float().to(device)

for i in range(len(GRID_list)):
    GRID_list[i-1] = torch.from_numpy(GRID_list[i-1]).float().to(device)
```

Loss functions

How accurate are cell distributions compared to training?

Whether the parameters for the new cell locations are actual signals?

Whether gene to pixel transformations are learned correctly?

Whether cell types are aligned with a reference?

Whether pixel image prediction is aligned with reality on a local predicted imaging spot level?

Whether pixel image prediction is aligned with reality on a global predicted imaging tissue level?

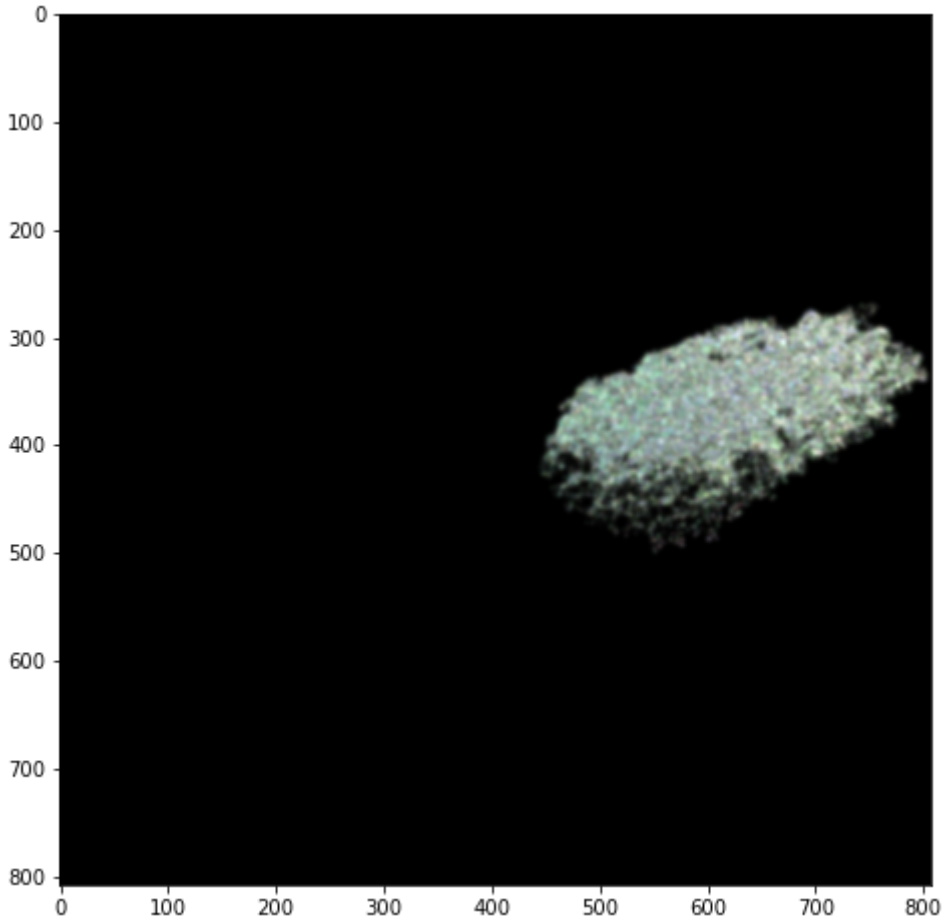
Whether the movement of cells around fits well with the idea of Waddington's marble analogy?

In [13]:
```
plt.rcParams['figure.figsize'] = (8, 8)

import time
import gc

start = time.time()

for epoch in range(10):  # loop over the dataset multiple times

    optimizer.zero_grad()
    grid_return_list, ST_list, spatial_gex_list = net(x_list, gex_list,pixel



# Initialise loss


    loss = torch.zeros(1).to(device)



    x1 = grid_return_list[0][0]
```

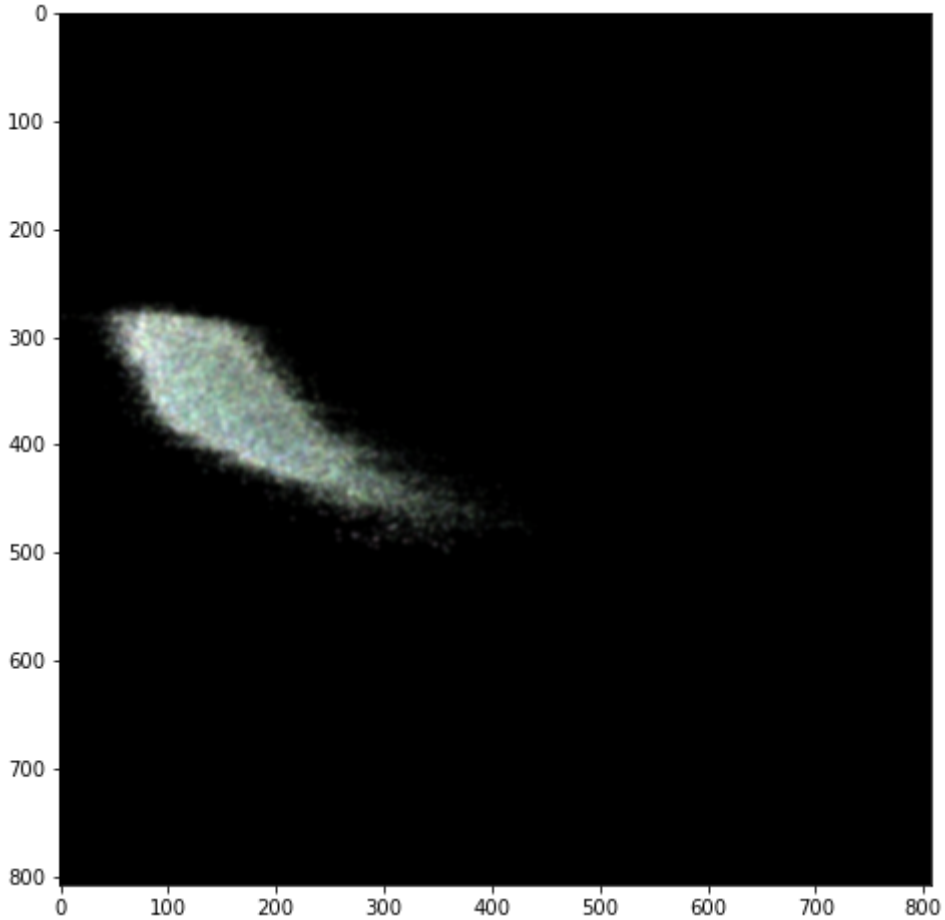
```
    x2 = grid_return_list[0][1]
    for i in range(1,len(grid_return_list)):



# Loss function questions how accurate are cell distributions compared to tr


        loss += mseloss(grid_return_list[i][0][:,:2],GRID_list[i-1])

        x1 = torch.vstack((x1,grid_return_list[i][0]))
        x2 = torch.vstack((x2,grid_return_list[i][1]))


# Loss function questions whether the parameters for the new cell locations

    loss += mseloss(x1,x2)



# Loss function questions whether gene to pixel transformations are learned


    for i in range(len(gex_list)):
        loss += mseloss(gex_list[i],spatial_gex_list[i][0])+mseloss(gex_list



# Loss function questions whether pixel image prediction is aligned with rea
# local predicted imaging spot level?


    for i in range(len(pixel_list)):
        loss += mseloss(pixel_list[i],ST_list[i+1][0])




    kde_grid_new = []
    cell_grid_new = []
    cell_prob_new = []
    for i in range(len(grid_return_list)):
        kde_grid_new += [[(to_kde_grid(M_list[i],(grid_return_list[i][0][:,:
        x_list[i] = 0.8*x_list[i] + 0.2*to_samples(kde_grid_new[i][0],(grid_



# Loss function questions whether pixel image prediction is aligned with rea
# global predicted imaging tissue level?

        if i>0:

            main_histology = np.max((kde_grid_new[i][0].reshape([(main_axis_
            main_histology = (main_histology-np.min(main_histology))/np.max(

            loss += mseloss(torch.from_numpy(main_histology).float().to(devi

        if i == 0 and epoch%5==4:

            x = (to_kde_grid(M_list[i],(grid_return_list[i][0][:,[0,1]].cpu(
            y = (to_kde_grid(M_list[i],(grid_return_list[i][0][:,[0,2]].cpu(
            z = (to_kde_grid(M_list[i],(grid_return_list[i][0][:,[1,2]].cpu(

            image_x = np.max((x.reshape([(main_axis_size+4*window_size)*(mai
```

```
            image_x = (image_x-np.min(image_x))/np.max(image_x-np.min(image_

            image_y = np.max((y.reshape([(main_axis_size+4*window_size)*(mai
            image_y = (image_y-np.min(image_y))/np.max(image_y-np.min(image_

            image_z = np.max((z.reshape([(main_axis_size+4*window_size)*(mai
            image_z = (image_z-np.min(image_z))/np.max(image_z-np.min(image_

            print("In-Silico Histology Examples - X-axis")
            plt.imshow(image_x)
            plt.show()

            print("In-Silico Histology Examples - Y-axis")
            plt.imshow(image_y)
            plt.show()

            print("In-Silico Histology Examples - Z-axis")
            plt.imshow(image_z)
            plt.show()

        else:



# Loss function questions whether the movement of cells around fits well wit


            loss += mseloss(x_list[i],to_samples(kde_grid_new[i][0],(grid_re



    del grid_return_list, ST_list, spatial_gex_list
    gc.collect()

    loss.backward()
    optimizer.step()

    end = time.time()
    if epoch%5==4:
        print([epoch,end - start,loss])

print('Finished Training')
```

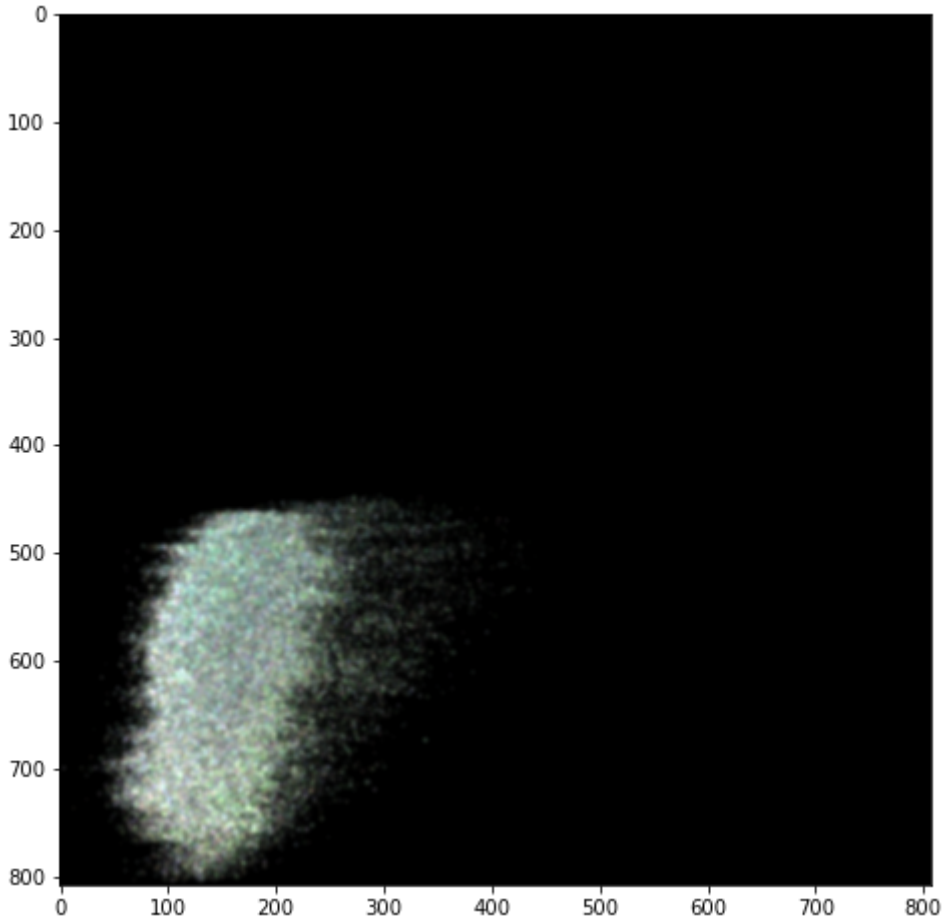
```
In-Silico Histology Examples - X-axis
```



In-Silico Histology Examples - Y-axis

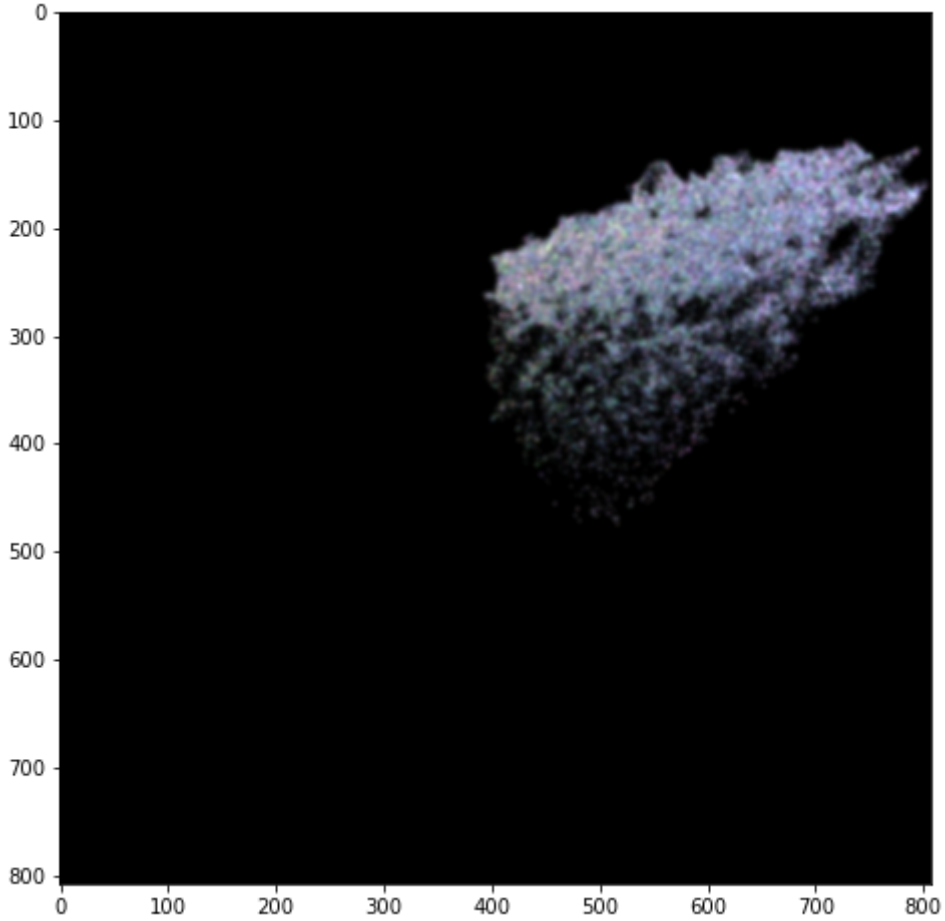


In-Silico Histology Examples - Z-axis

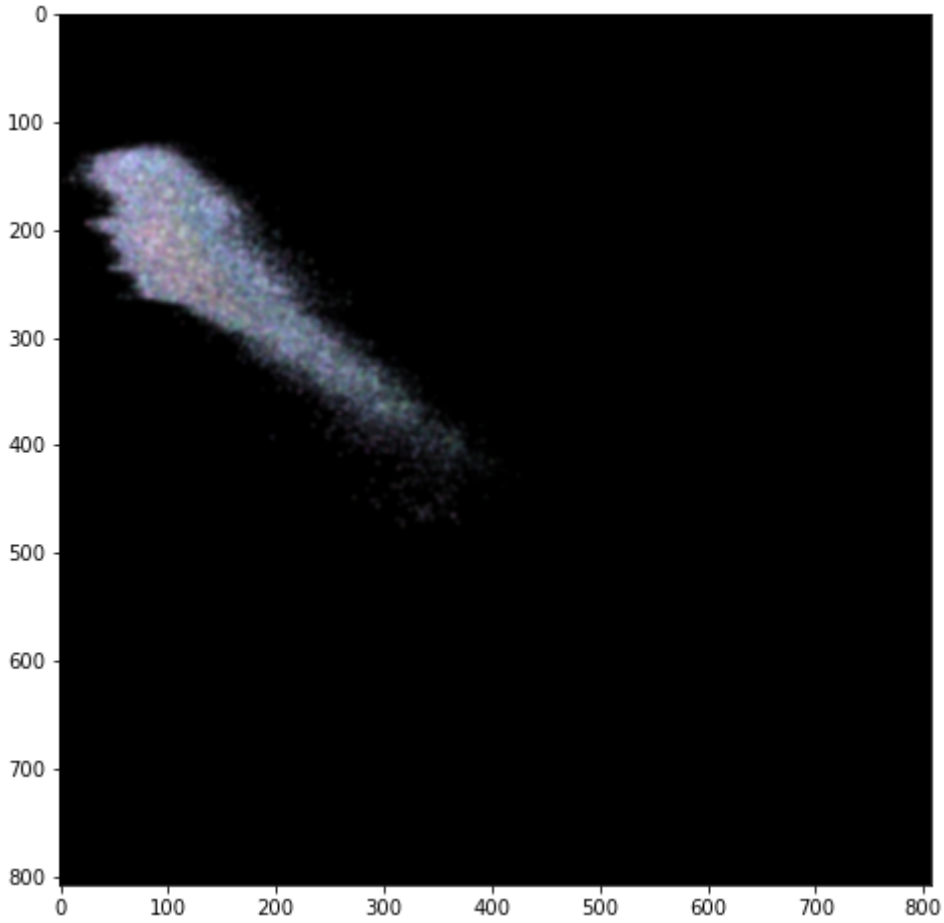



```
[4, 681.13592004776, tensor([54675000.], grad_fn=<AddBackward0>)]
```

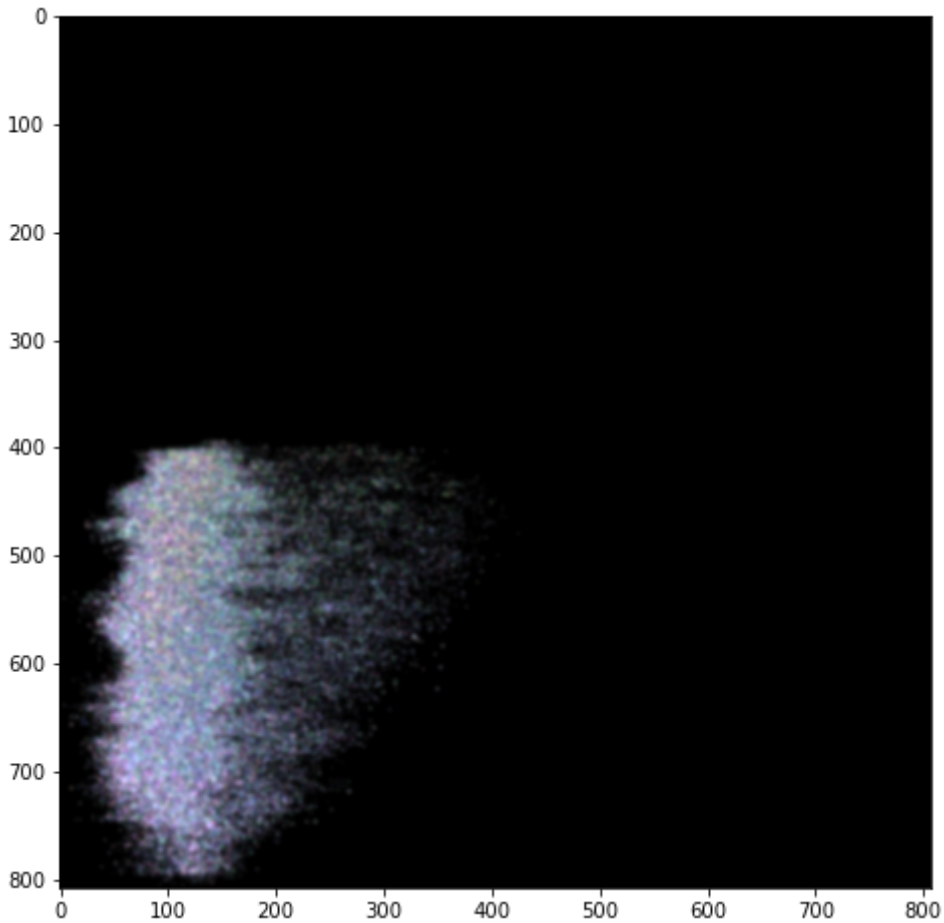
In-Silico Histology Examples - X-axis



In-Silico Histology Examples - Y-axis




```
In-Silico Histology Examples - Z-axis
```




```
[9, 1362.5275220870972, tensor([17450640.], grad_fn=<AddBackward0>)]
Finished Training
```

In [14]:
```
PATH1 = "/home/jupyter/lmform_analysis/base_cnn/state_dict_VisualForm.pt"
PATH2 = "/home/jupyter/lmform_analysis/base_cnn/to_train_VisualForm.pt"
```

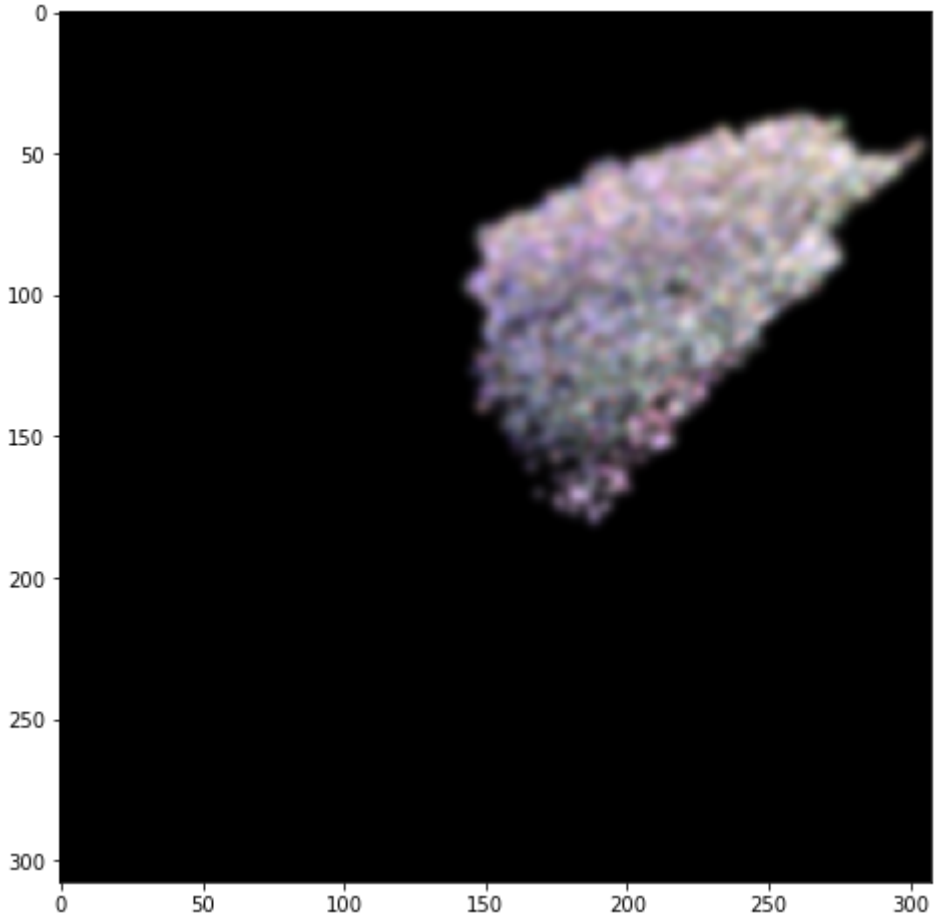
```
PATH3 = "/home/jupyter/lmform_analysis/base_cnn/all_VisualForm.pt"

torch.save(net.state_dict(), PATH1)
torch.save({
            'epoch': epoch,
            'model_state_dict': net.state_dict(),
            'optimizer_state_dict': optimizer.state_dict(),
            'loss': loss
}, PATH2)
torch.save(net, PATH3)
```

In [47]:
```
grid_return_list, ST_list, spatial_gex_list = net(x_list,gex_list,pixel_list


for i in range(len(GRID_list)):
    grid_return_list[i+1][0][:,:2] = GRID_list[i]

main_axis_size = 300

kde_grid_new_x = []
kde_grid_new_y = []
kde_grid_new_z = []


for i in range(len(grid_return_list)):
    kde_grid_new_x += [[to_kde_grid(M_list[i],grid_return_list[i][0].cpu().d
    kde_grid_new_y += [[to_kde_grid(M_list[i],grid_return_list[i][0].cpu().d
    kde_grid_new_z += [[to_kde_grid(M_list[i],grid_return_list[i][0].cpu().d
```

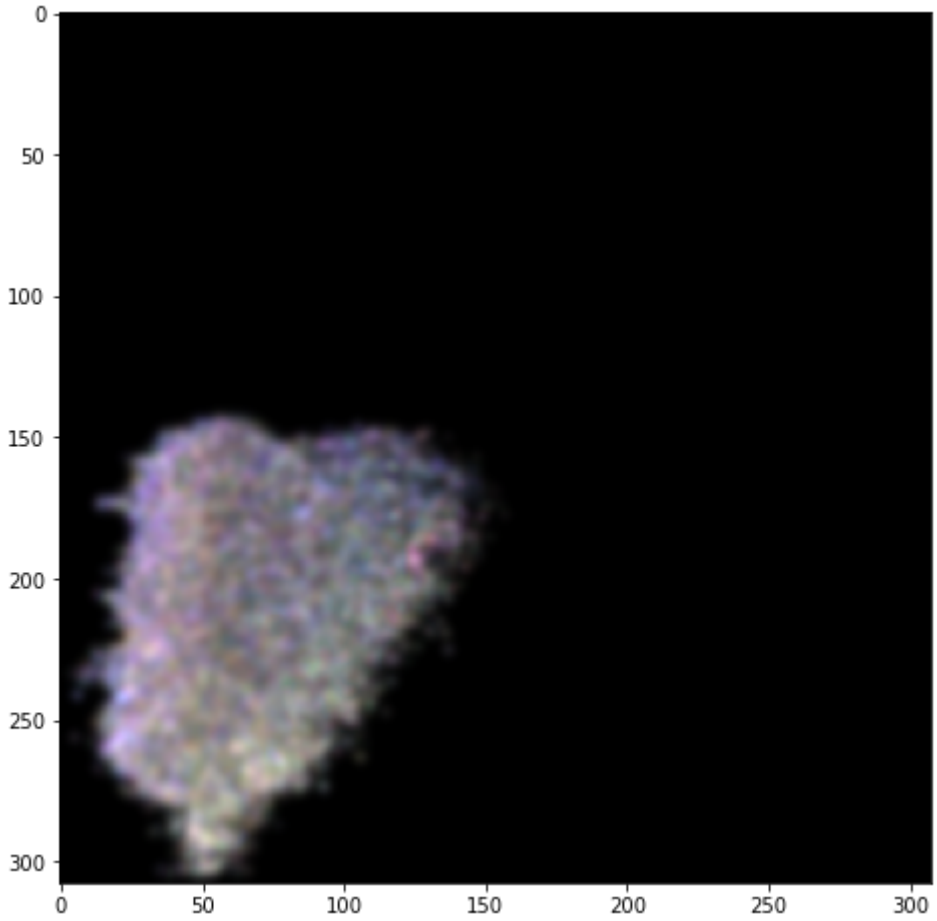
In [48]:
```
plt.rcParams['figure.figsize'] = (8, 8)


for i in range(5):

    x = kde_grid_new_x[i][0]
    y = kde_grid_new_y[i][0]
    z = kde_grid_new_z[i][0]

    image_x = np.max((x.reshape([(main_axis_size+4*window_size)*(main_axis_s
    image_x = (image_x-np.min(image_x))/np.max(image_x-np.min(image_x))

    image_y = np.max((y.reshape([(main_axis_size+4*window_size)*(main_axis_s
    image_y = (image_y-np.min(image_y))/np.max(image_y-np.min(image_y))

    image_z = np.max((z.reshape([(main_axis_size+4*window_size)*(main_axis_s
    image_z = (image_z-np.min(image_z))/np.max(image_z-np.min(image_z))


    if i == 0:
        print("In-Silico Histology Examples")

    if i > 0:
        print("Real Histology Examples")


    print("X-axis")
    plt.imshow(image_x)
    plt.show()

    print("Y-axis")
    plt.imshow(image_y)
    plt.show()
```

```
    print("Z-axis")
    plt.imshow(image_z)
    plt.show()
```

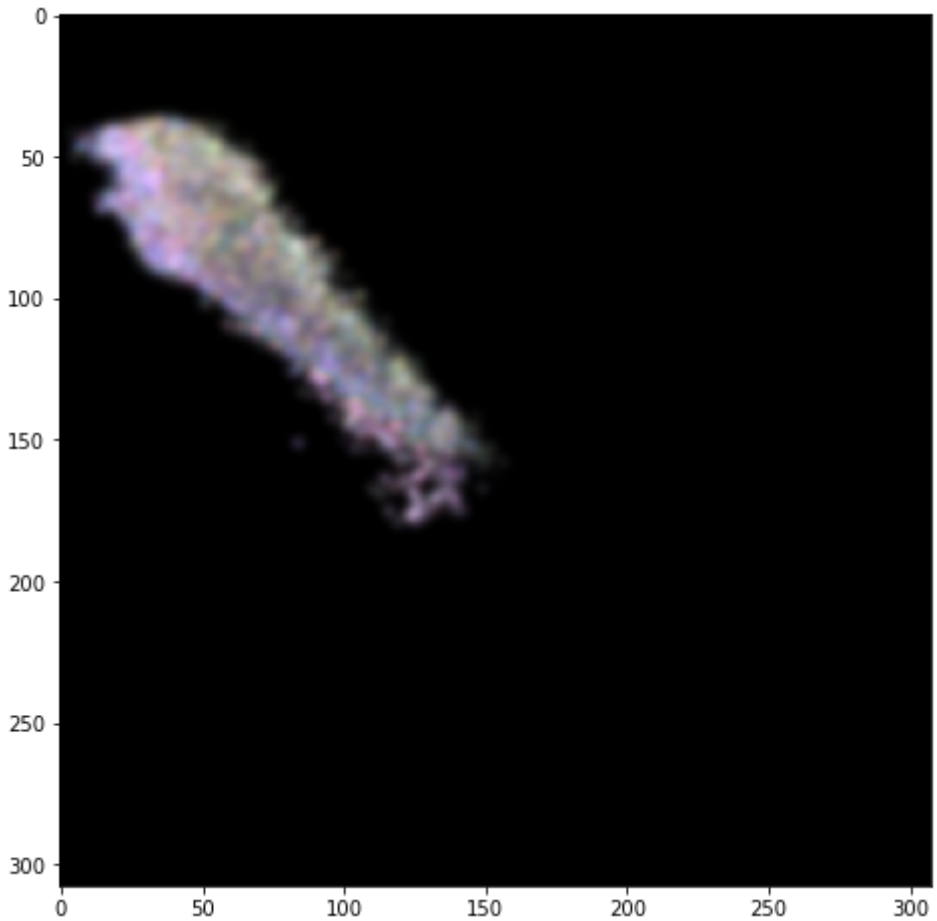
```
In-Silico Histology Examples
X-axis
```



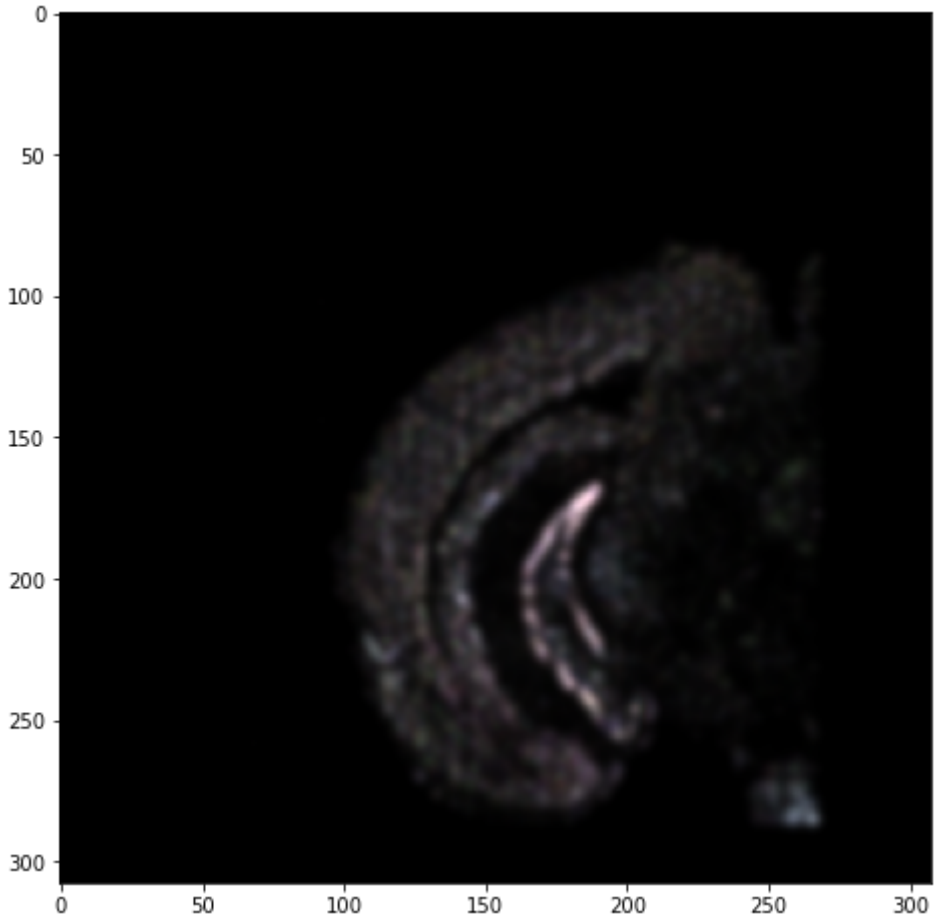

```
Y-axis
```



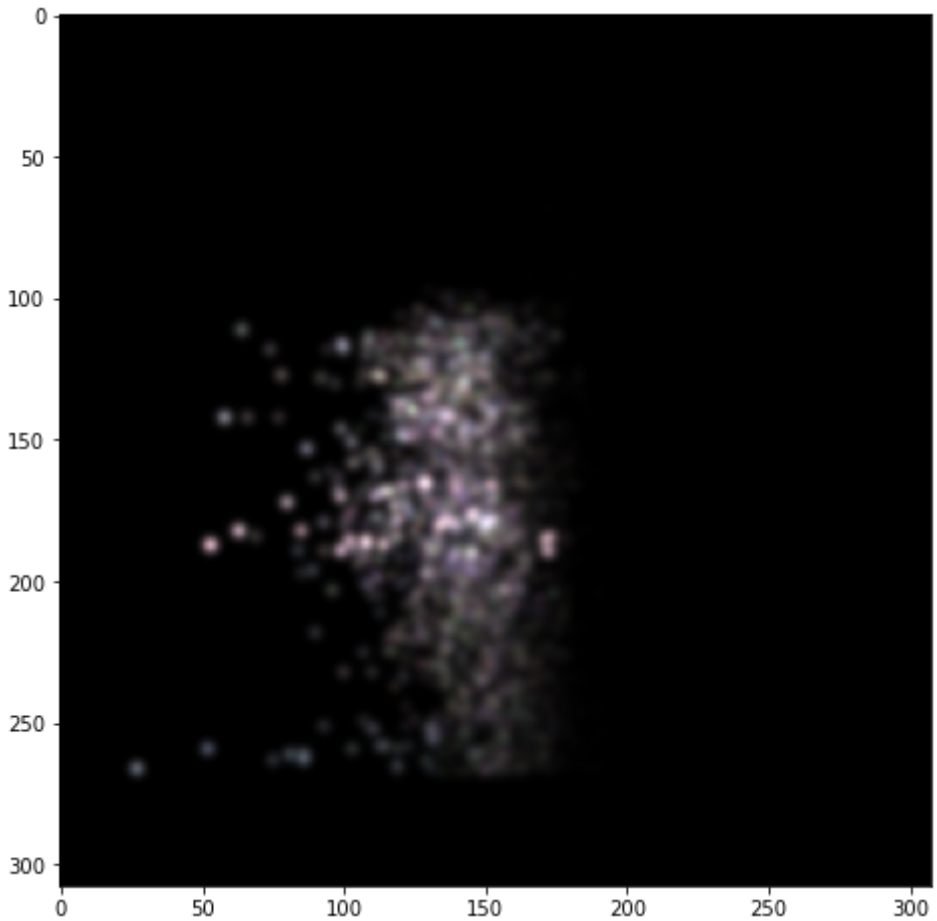
Z-axis



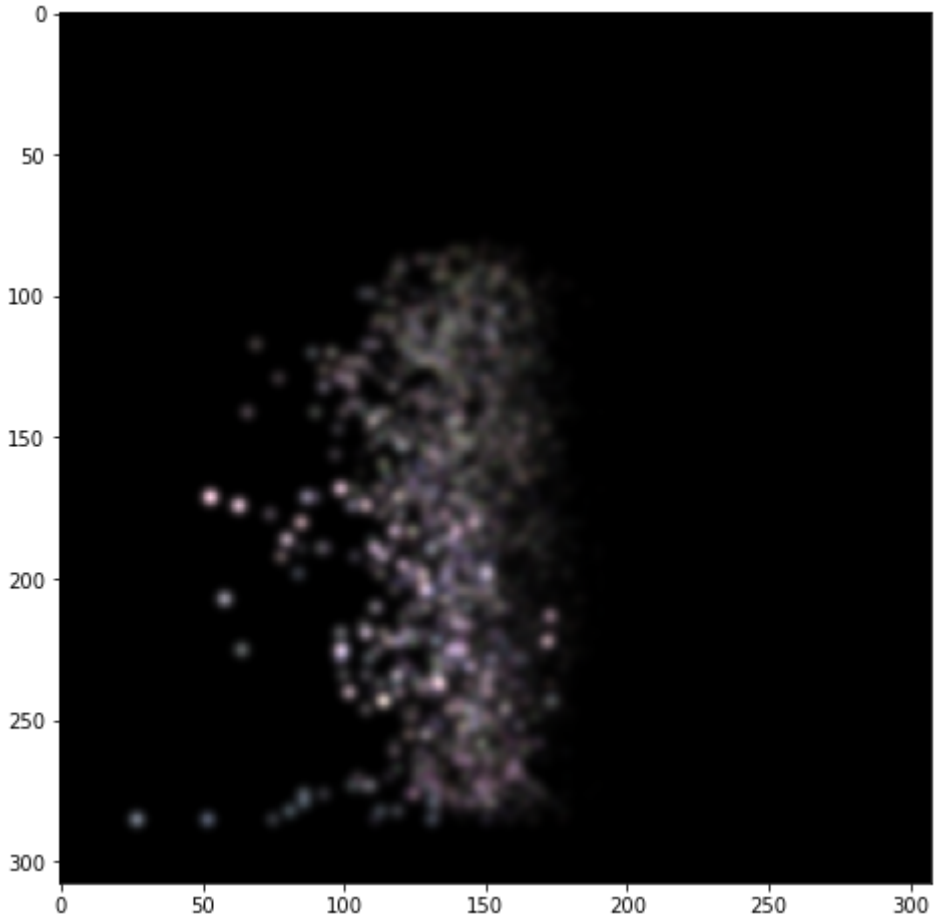
Real Histology Examples
X-axis


0
50
100
150
200
250
300
0
50
100
150
200
250
300

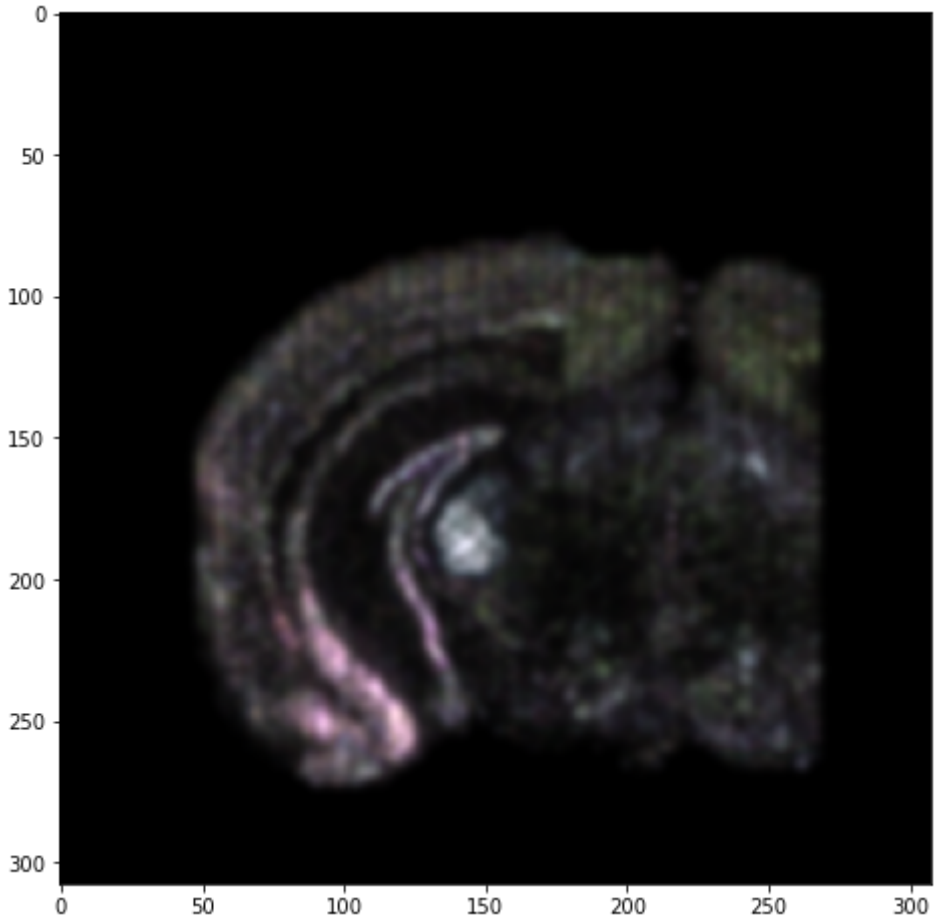
Y-axis


0
50
100
150
200
250
300
0
50
100
150
200
250
300

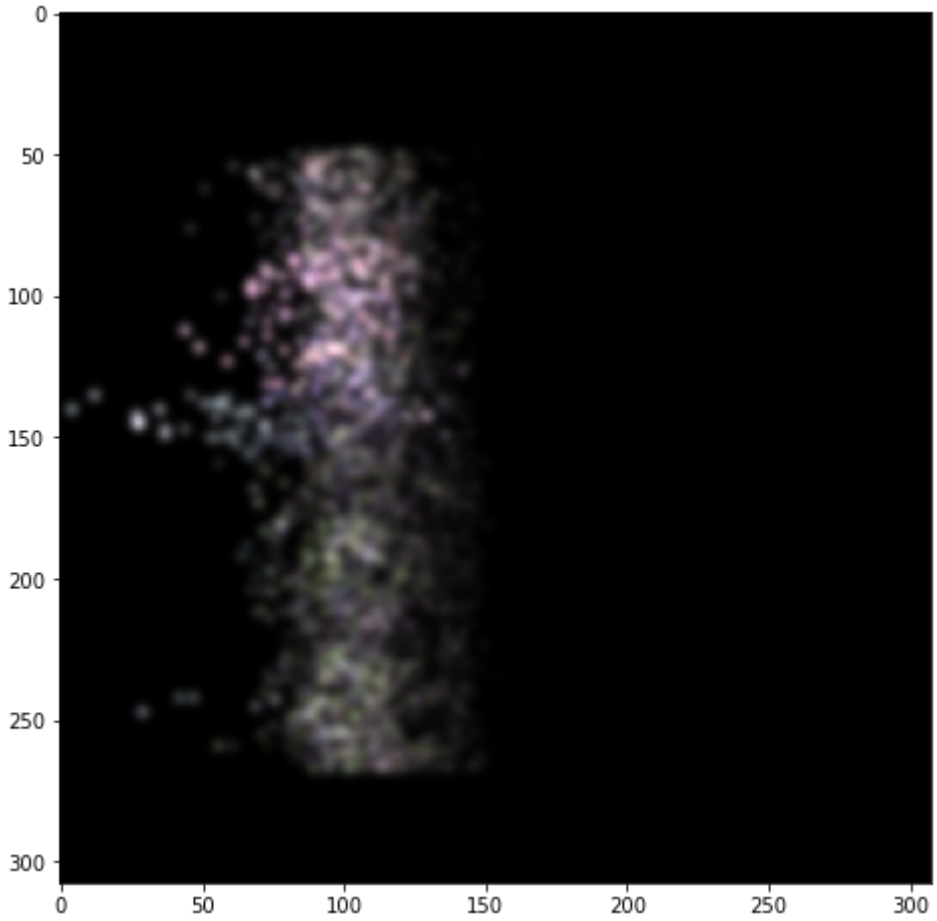
Z-axis



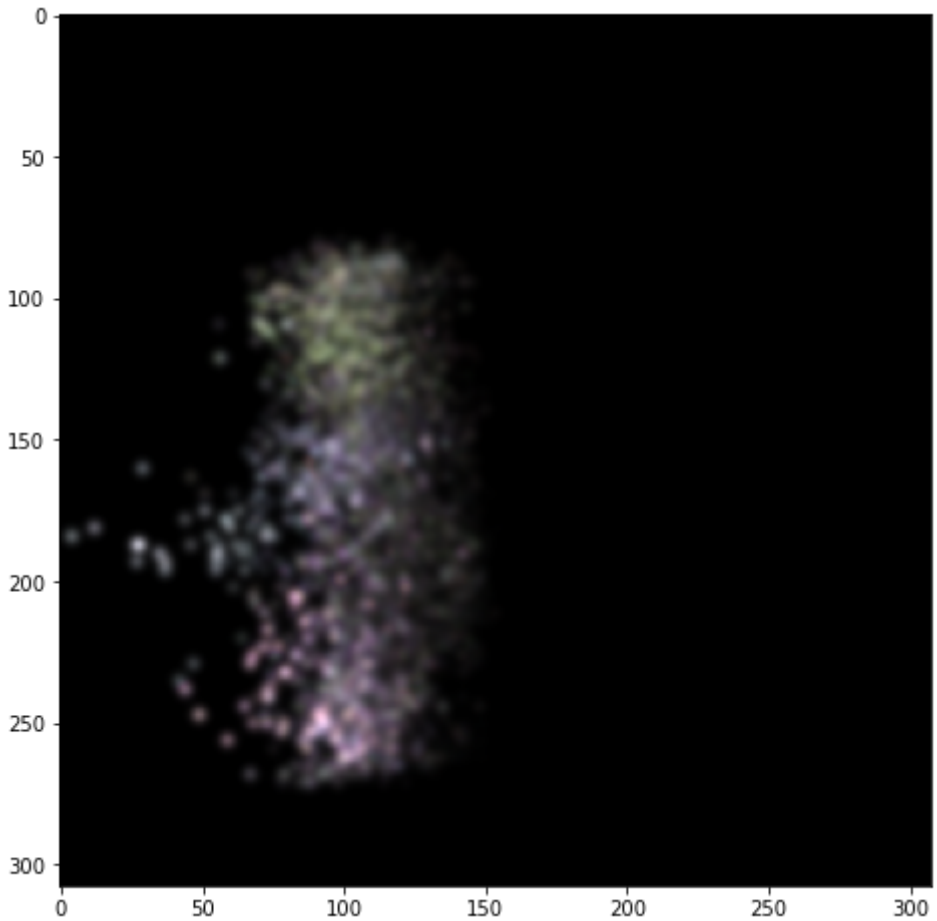
Real Histology Examples
X-axis



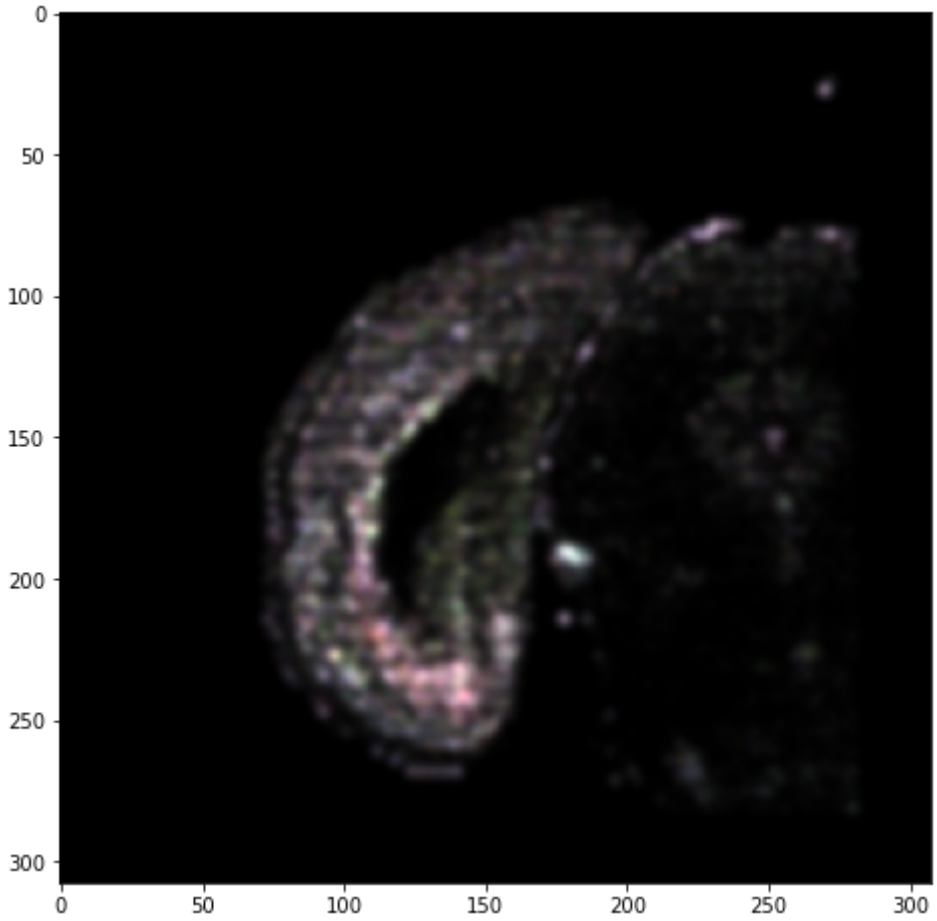
Y-axis



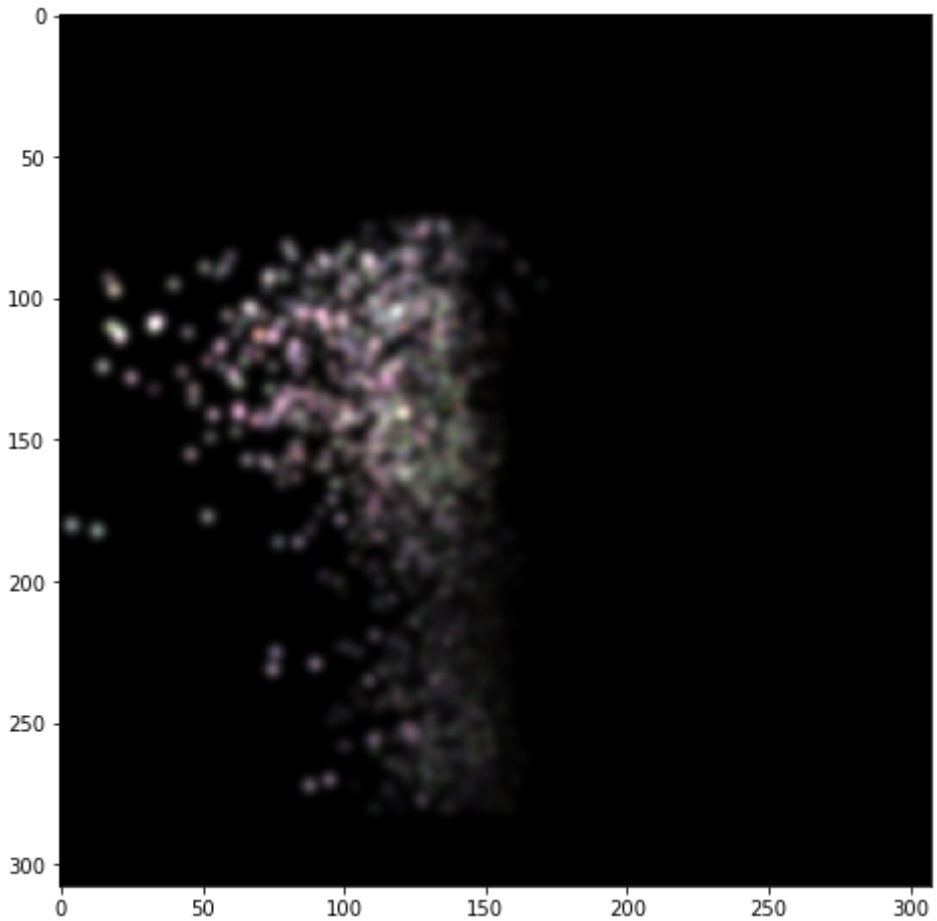
Z-axis



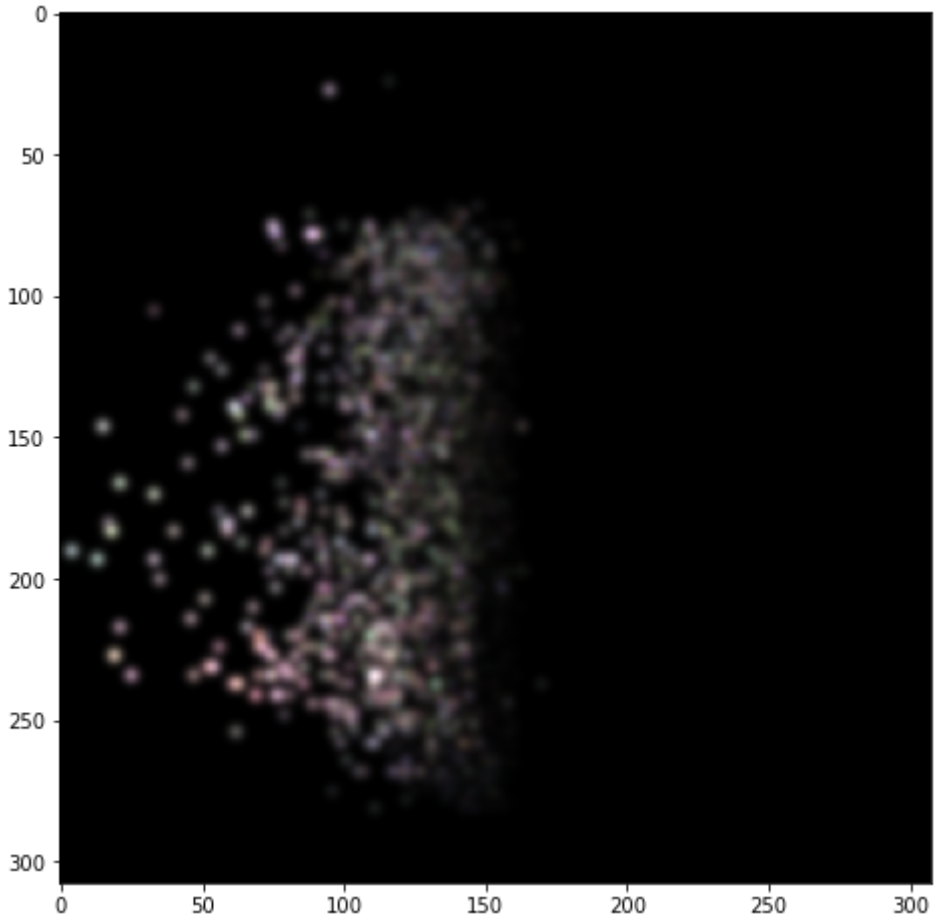
Real Histology Examples
X-axis


0
50
100
150
200
250
300
0
50
100
150
200
250
300

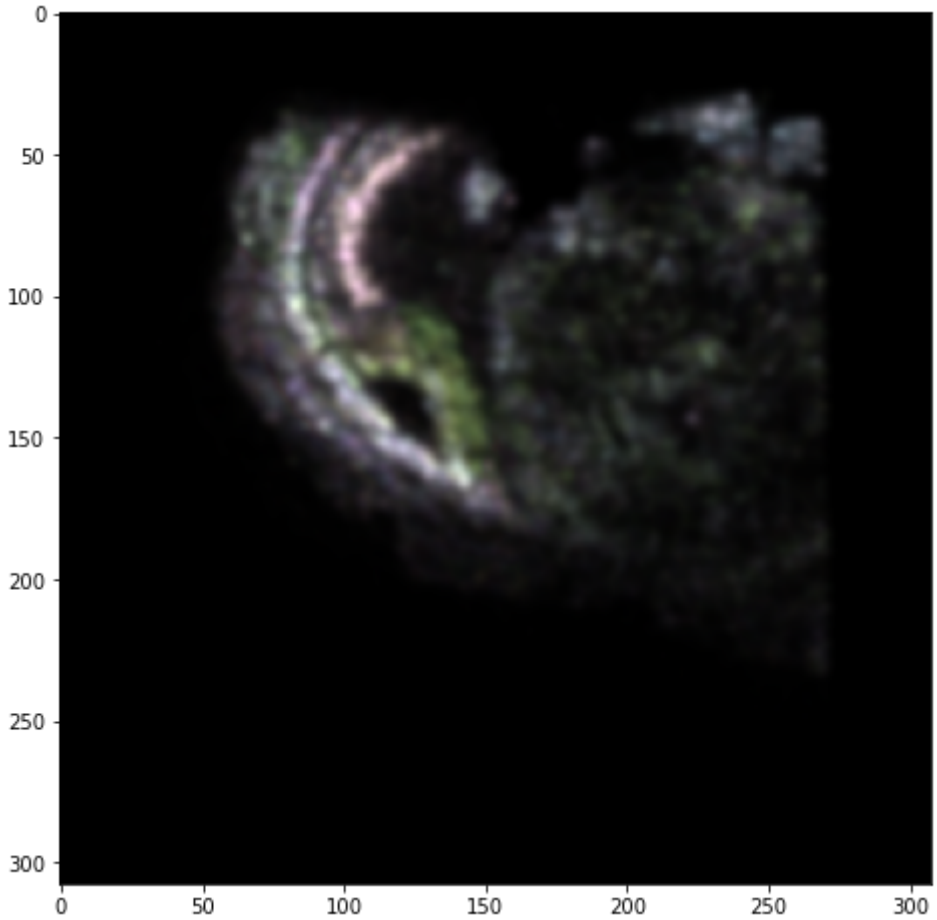
Y-axis


0
50
100
150
200
250
300
0
50
100
150
200
250
300

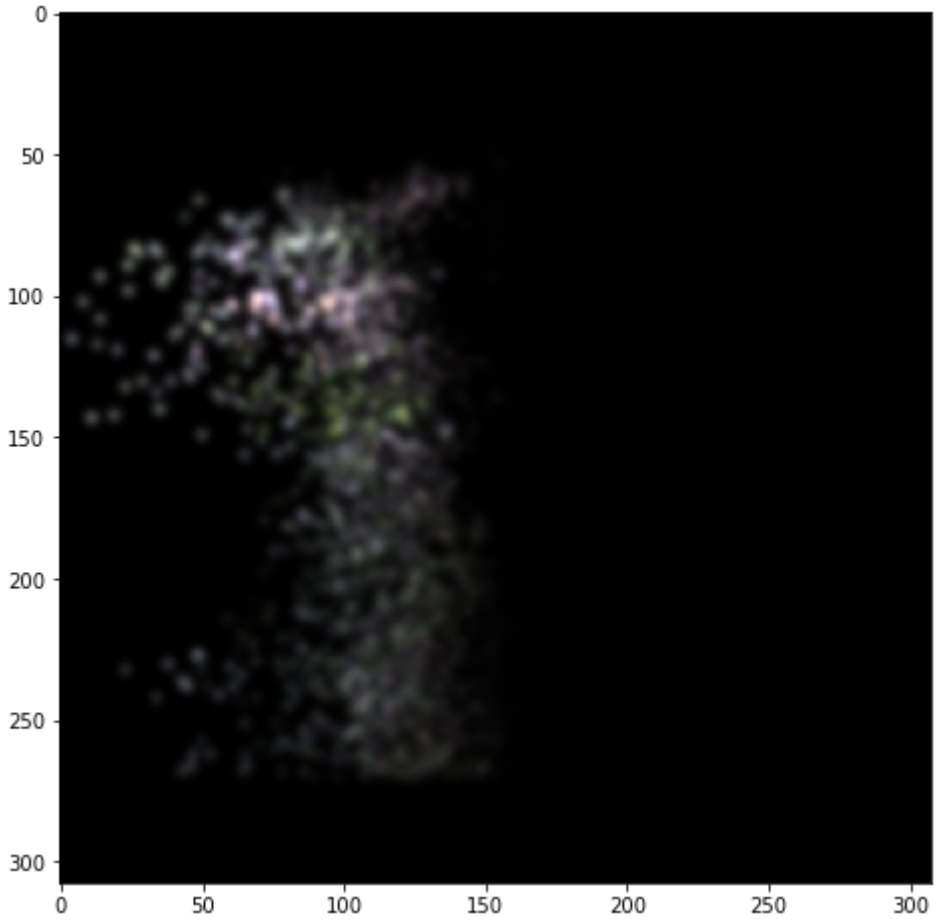
Z-axis



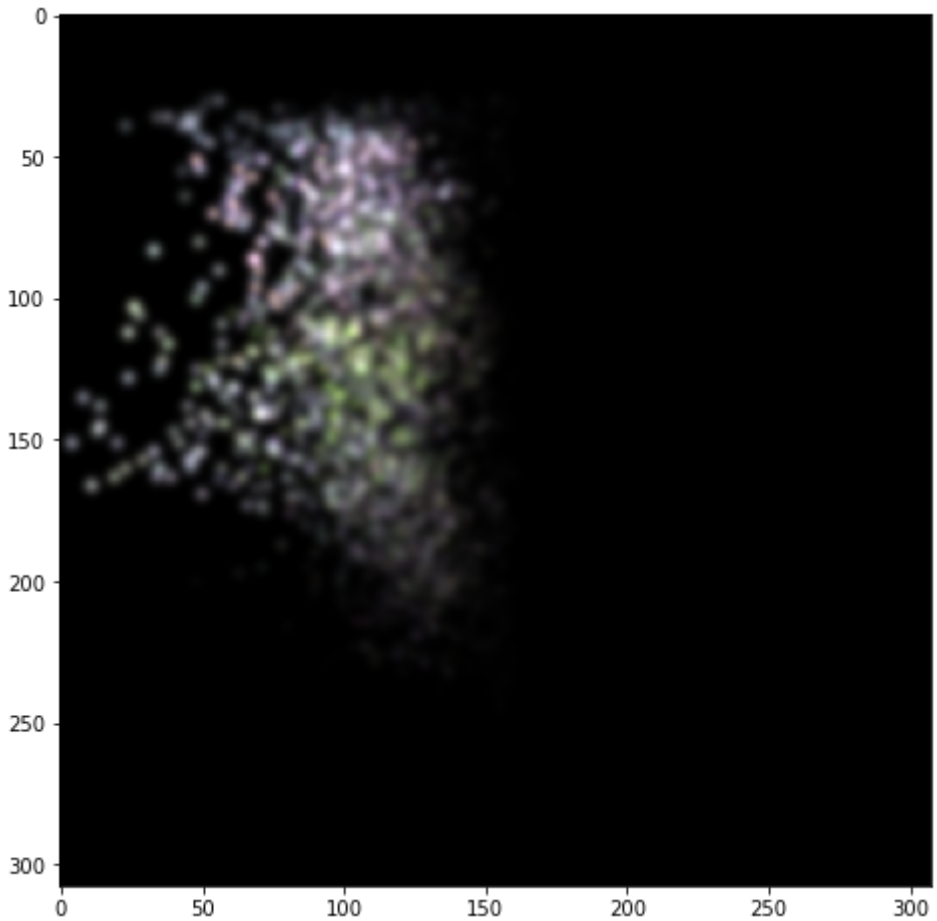
Real Histology Examples
X-axis



Y-axis



Z-axis

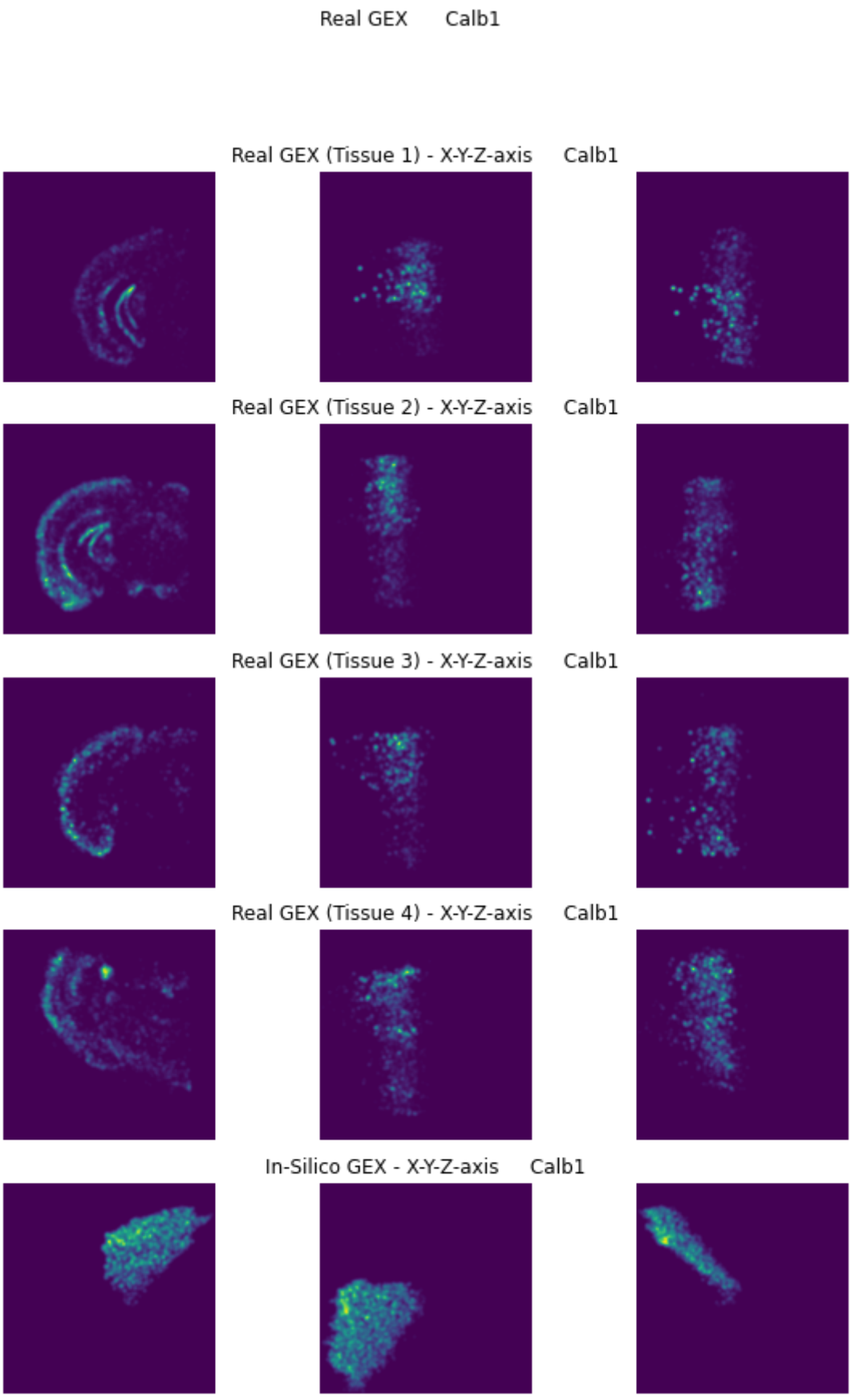


In [74]:
```
print("Plotting in-silico ST data via loss function optimisation through lea

for i in range(len(genes)):
```

```
    plt.rcParams['figure.figsize'] = (10, 14)

    fig, axs = plt.subplots(5,3,rasterized=True)

    fig.suptitle("Real GEX      " + genes[i])

    axs[0,1].set_title("Real GEX (Tissue 1) - X-Y-Z-axis     " + genes[i])

    axs[0,0].imshow(kde_grid_new_x[1][0][:,:,i])
    axs[0,1].imshow(kde_grid_new_y[1][0][:,:,i])
    axs[0,2].imshow(kde_grid_new_z[1][0][:,:,i])

    axs[1,1].set_title("Real GEX (Tissue 2) - X-Y-Z-axis     " + genes[i])

    axs[1,0].imshow(kde_grid_new_x[2][0][:,:,i])
    axs[1,1].imshow(kde_grid_new_y[2][0][:,:,i])
    axs[1,2].imshow(kde_grid_new_z[2][0][:,:,i])

    axs[2,1].set_title("Real GEX (Tissue 3) - X-Y-Z-axis     " + genes[i])

    axs[2,0].imshow(kde_grid_new_x[3][0][:,:,i])
    axs[2,1].imshow(kde_grid_new_y[3][0][:,:,i])
    axs[2,2].imshow(kde_grid_new_z[3][0][:,:,i])

    axs[3,1].set_title("Real GEX (Tissue 4) - X-Y-Z-axis     " + genes[i])

    axs[3,0].imshow(kde_grid_new_x[4][0][:,:,i])
    axs[3,1].imshow(kde_grid_new_y[4][0][:,:,i])
    axs[3,2].imshow(kde_grid_new_z[4][0][:,:,i])

    axs[0,0].axis('off')
    axs[0,1].axis('off')
    axs[0,2].axis('off')

    axs[1,0].axis('off')
    axs[1,1].axis('off')
    axs[1,2].axis('off')

    axs[2,0].axis('off')
    axs[2,1].axis('off')
    axs[2,2].axis('off')

    axs[3,0].axis('off')
    axs[3,1].axis('off')
    axs[3,2].axis('off')

    axs[4,1].set_title("In-Silico GEX - X-Y-Z-axis     " + genes[i])

    axs[4,0].imshow(kde_grid_new_x[0][0][:,:,i])
    axs[4,1].imshow(kde_grid_new_y[0][0][:,:,i])
    axs[4,2].imshow(kde_grid_new_z[0][0][:,:,i])

    axs[4,0].axis('off')
    axs[4,1].axis('off')
    axs[4,2].axis('off')

    fig.savefig("./scMaps/mouse___"+genes[i]+'.png')
```

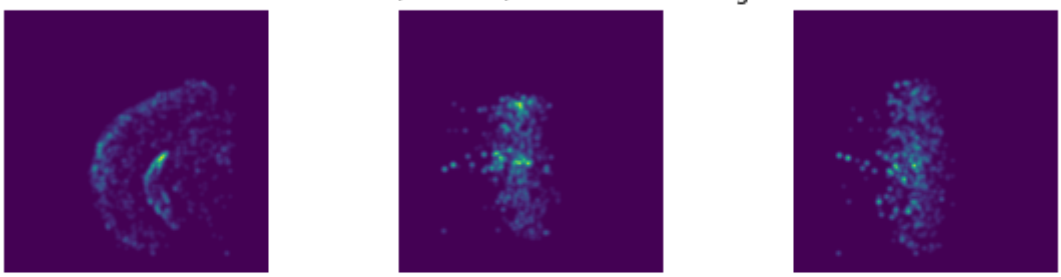
```
Plotting in-silico ST data via loss function optimisation through learning p
roper gene to pixel transformations and cell placement via KNN gene convolut
ions
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:8: RuntimeWarni
ng: More than 20 figures have been opened. Figures created through the pyplo
t interface (`matplotlib.pyplot.figure`) are retained until explicitly close
d and may consume too much memory. (To control this warning, see the rcParam
`figure.max_open_warning`).
```

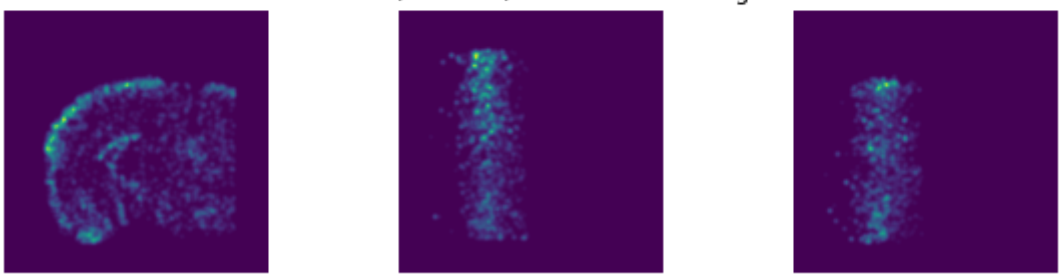


Real GEX Rasgrf2

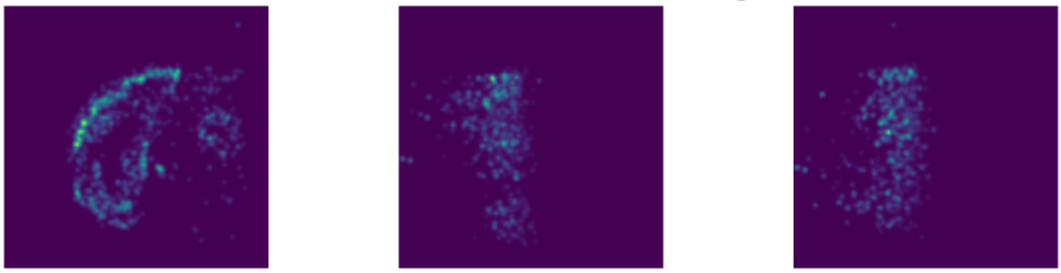
Real GEX (Tissue 1) - X-Y-Z-axis Rasgrf2



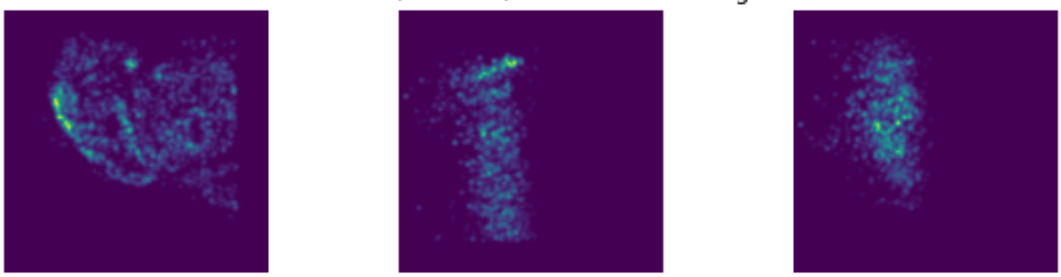
Real GEX (Tissue 2) - X-Y-Z-axis Rasgrf2



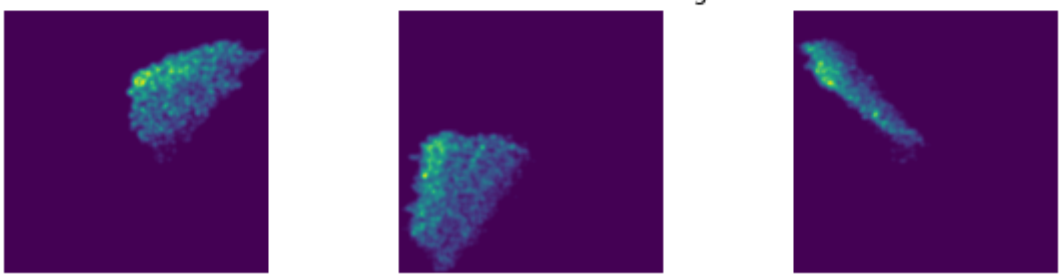
Real GEX (Tissue 3) - X-Y-Z-axis Rasgrf2



Real GEX (Tissue 4) - X-Y-Z-axis Rasgrf2

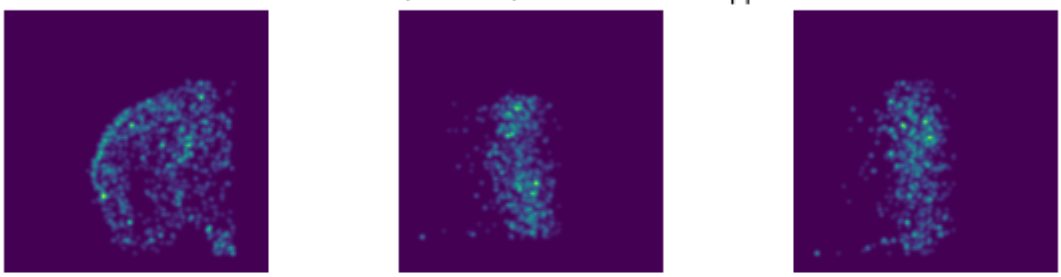


In-Silico GEX - X-Y-Z-axis Rasgrf2

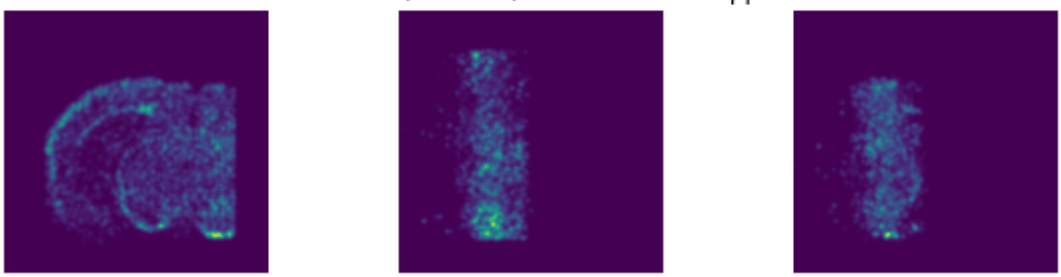


Real GEX Enpp2

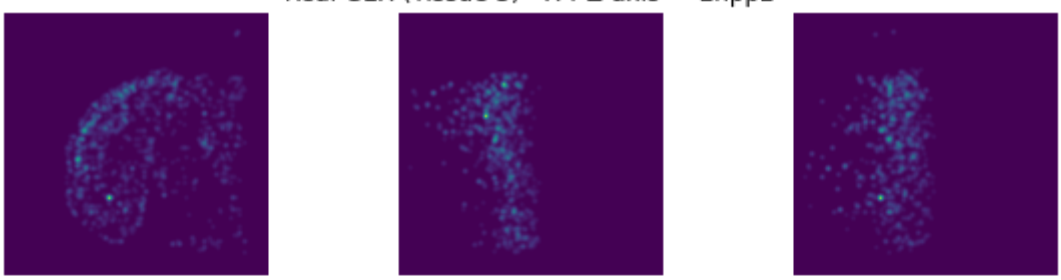
Real GEX (Tissue 1) - X-Y-Z-axis Enpp2



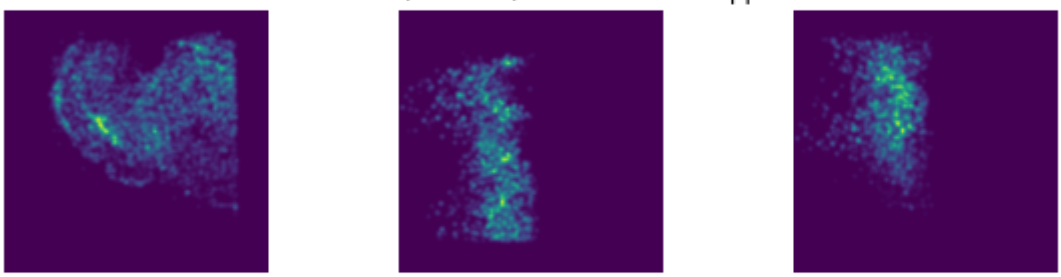
Real GEX (Tissue 2) - X-Y-Z-axis Enpp2



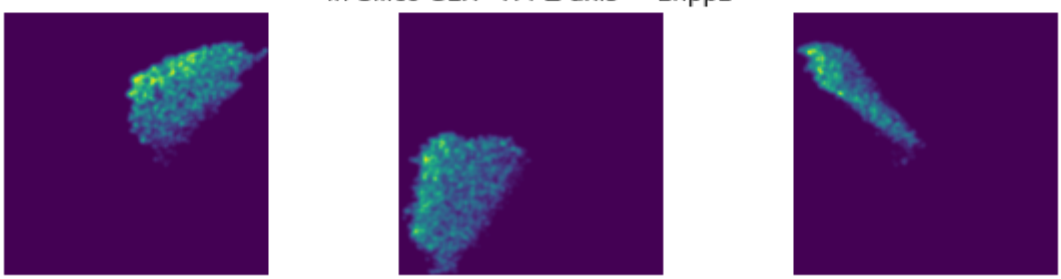
Real GEX (Tissue 3) - X-Y-Z-axis Enpp2



Real GEX (Tissue 4) - X-Y-Z-axis Enpp2

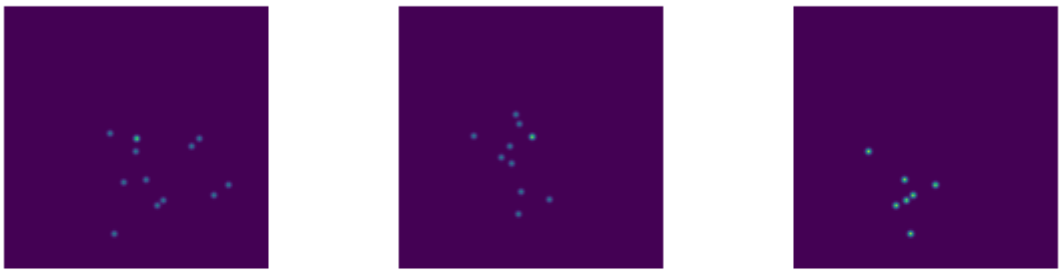


In-Silico GEX - X-Y-Z-axis Enpp2



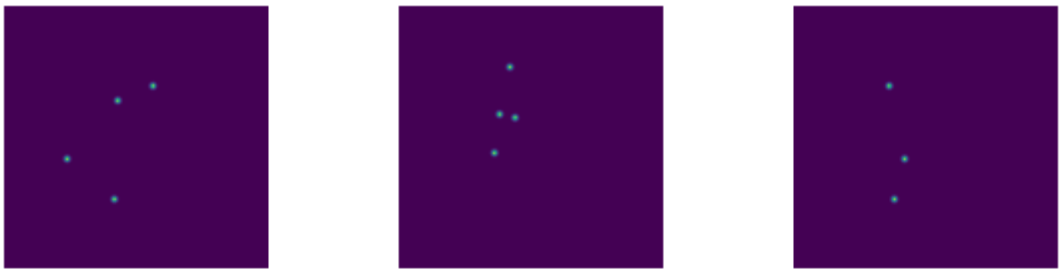
Real GEX Col19a1

Real GEX (Tissue 1) - X-Y-Z-axis Col19a1

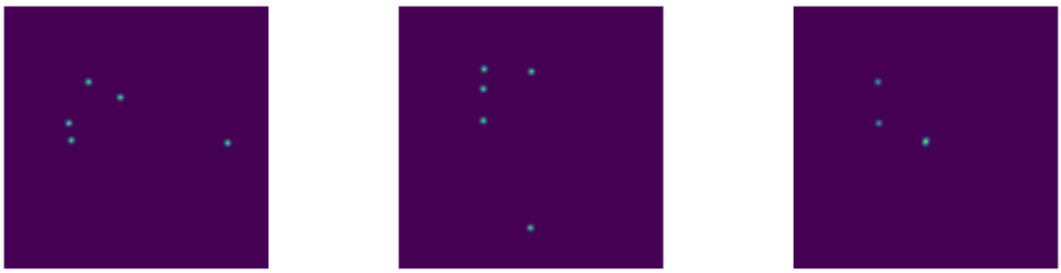


Real GEX (Tissue 2) - X-Y-Z-axis Col19a1

Real GEX (Tissue 3) - X-Y-Z-axis Col19a1



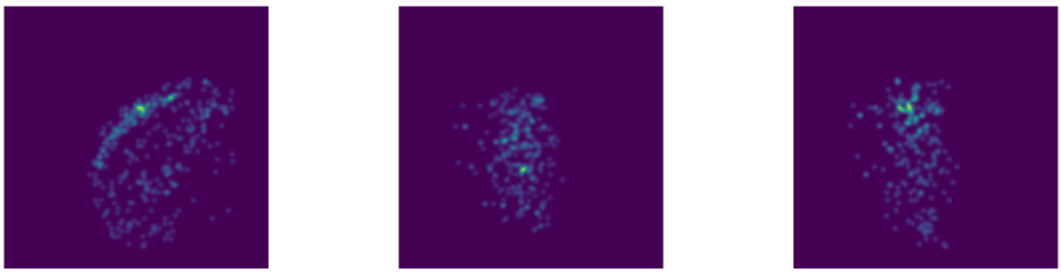
Real GEX (Tissue 4) - X-Y-Z-axis Col19a1



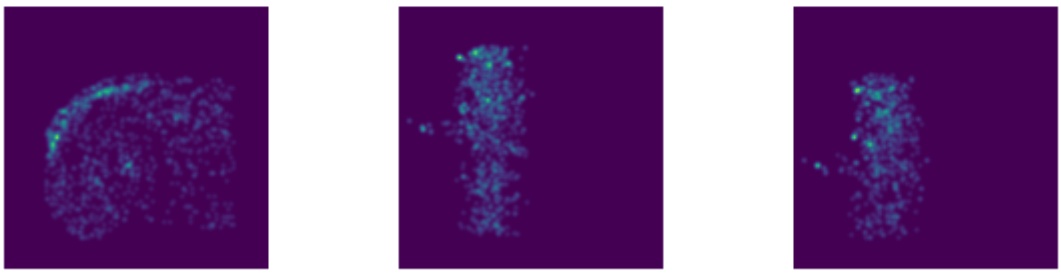
In-Silico GEX - X-Y-Z-axis Col19a1

Real GEX Rorb

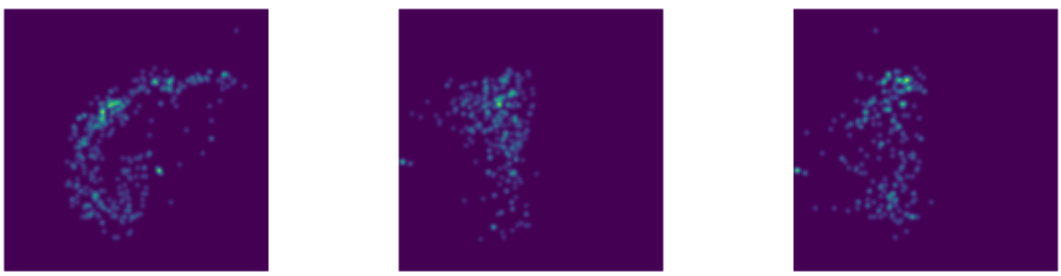
Real GEX (Tissue 1) - X-Y-Z-axis Rorb



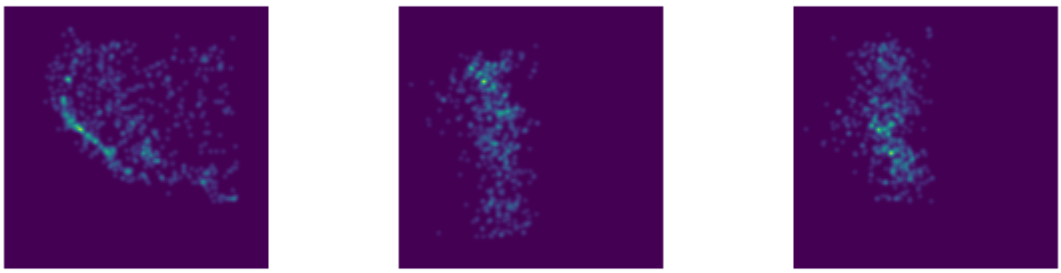
Real GEX (Tissue 2) - X-Y-Z-axis Rorb



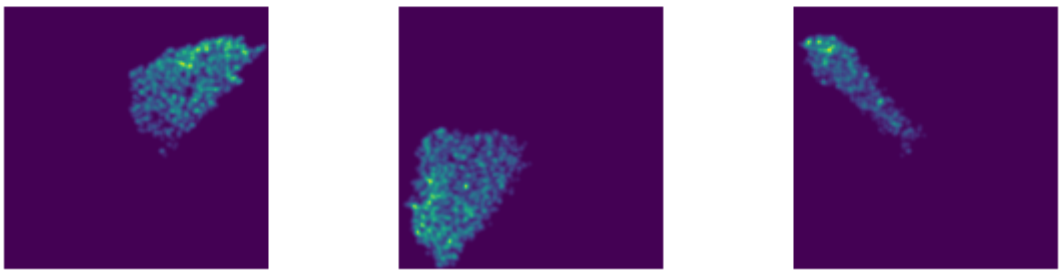
Real GEX (Tissue 3) - X-Y-Z-axis Rorb



Real GEX (Tissue 4) - X-Y-Z-axis Rorb

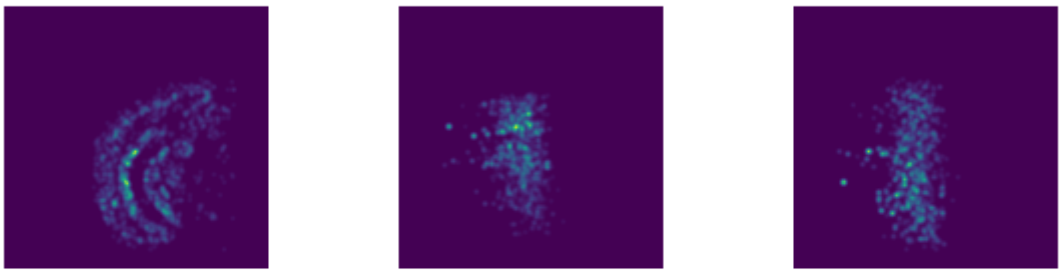


In-Silico GEX - X-Y-Z-axis Rorb

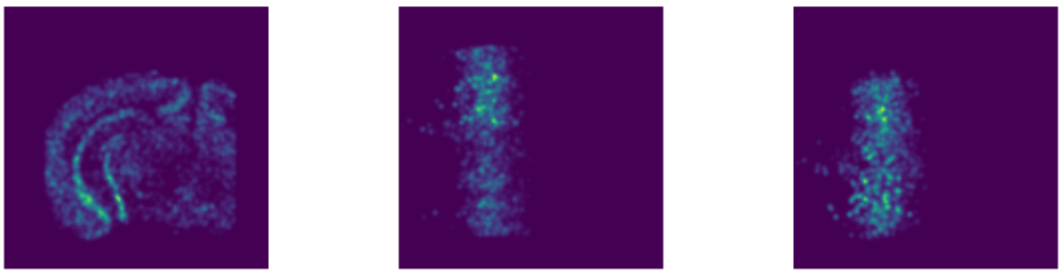


Real GEX Slc24a3

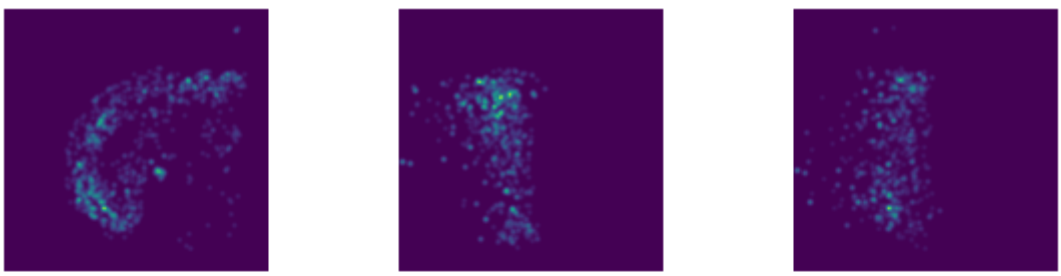
Real GEX (Tissue 1) - X-Y-Z-axis Slc24a3



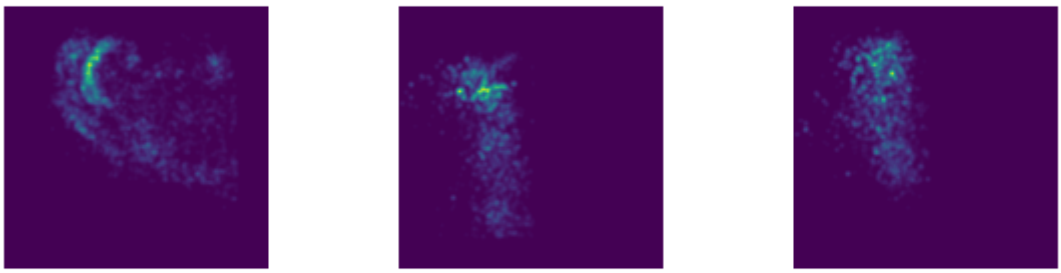
Real GEX (Tissue 2) - X-Y-Z-axis Slc24a3



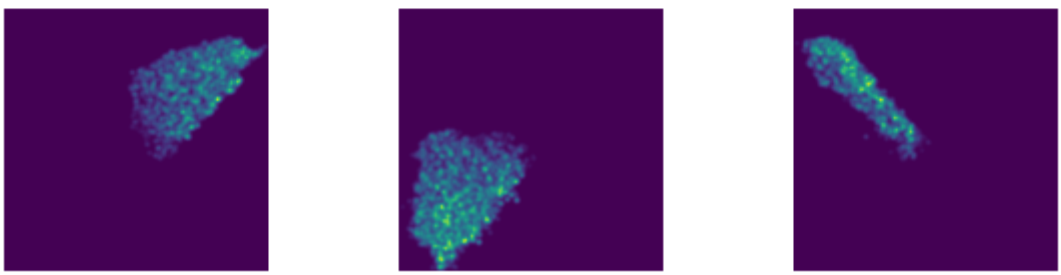
Real GEX (Tissue 3) - X-Y-Z-axis Slc24a3



Real GEX (Tissue 4) - X-Y-Z-axis Slc24a3

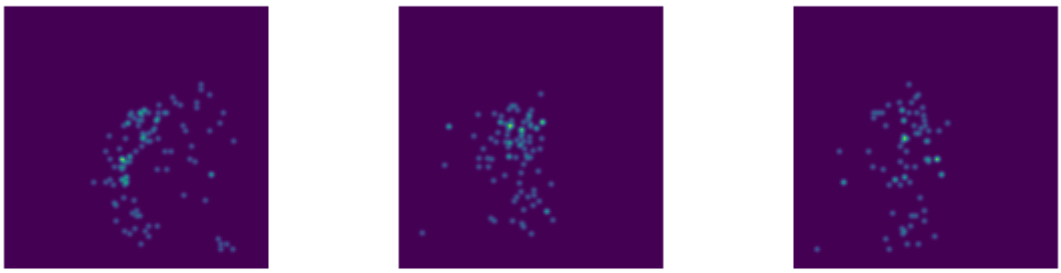


In-Silico GEX - X-Y-Z-axis Slc24a3

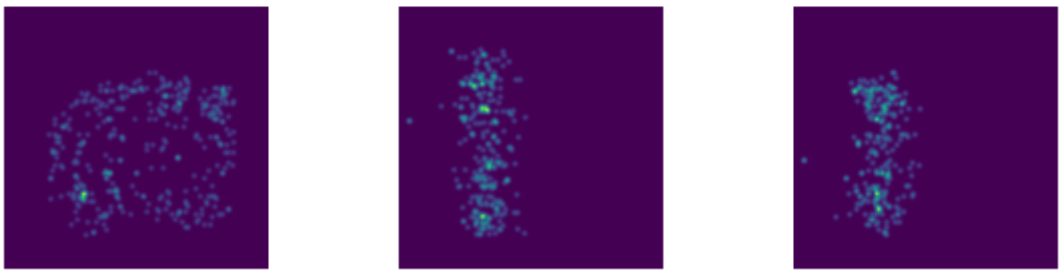


Real GEX Galntl6

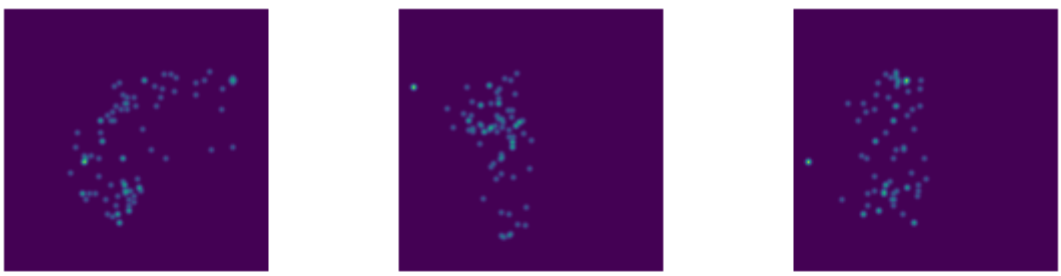
Real GEX (Tissue 1) - X-Y-Z-axis Galntl6



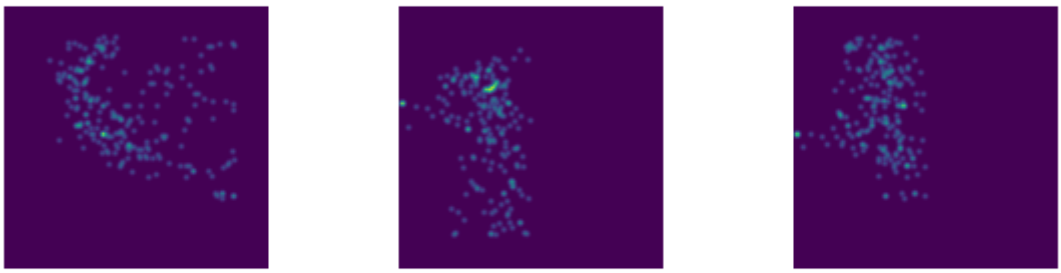
Real GEX (Tissue 2) - X-Y-Z-axis Galntl6



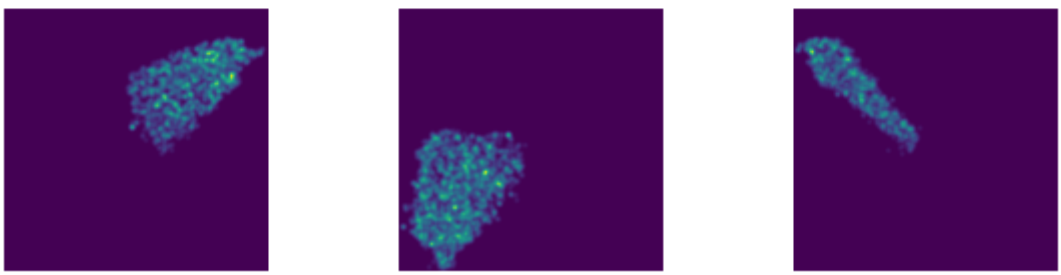
Real GEX (Tissue 3) - X-Y-Z-axis Galntl6



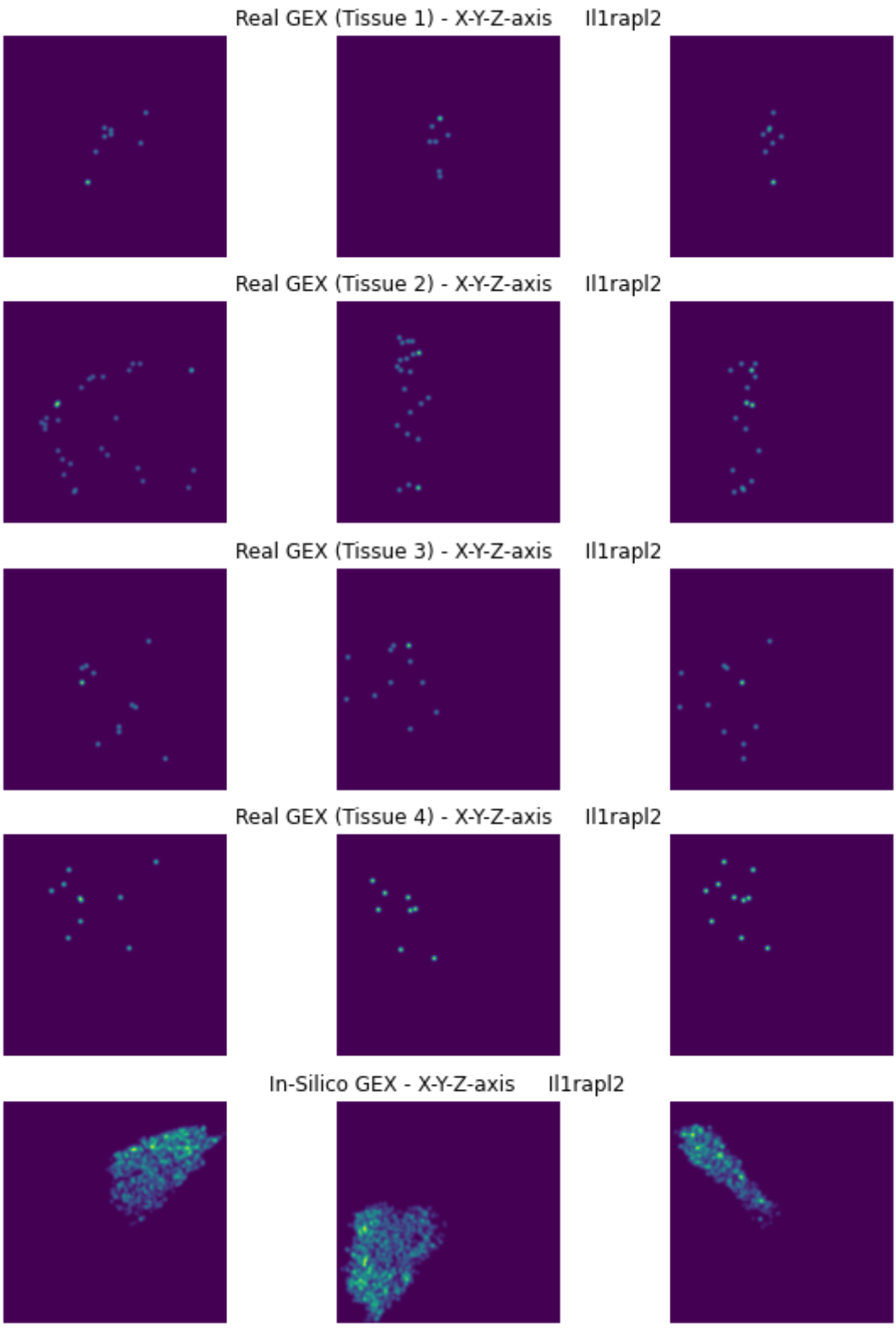
Real GEX (Tissue 4) - X-Y-Z-axis Galntl6



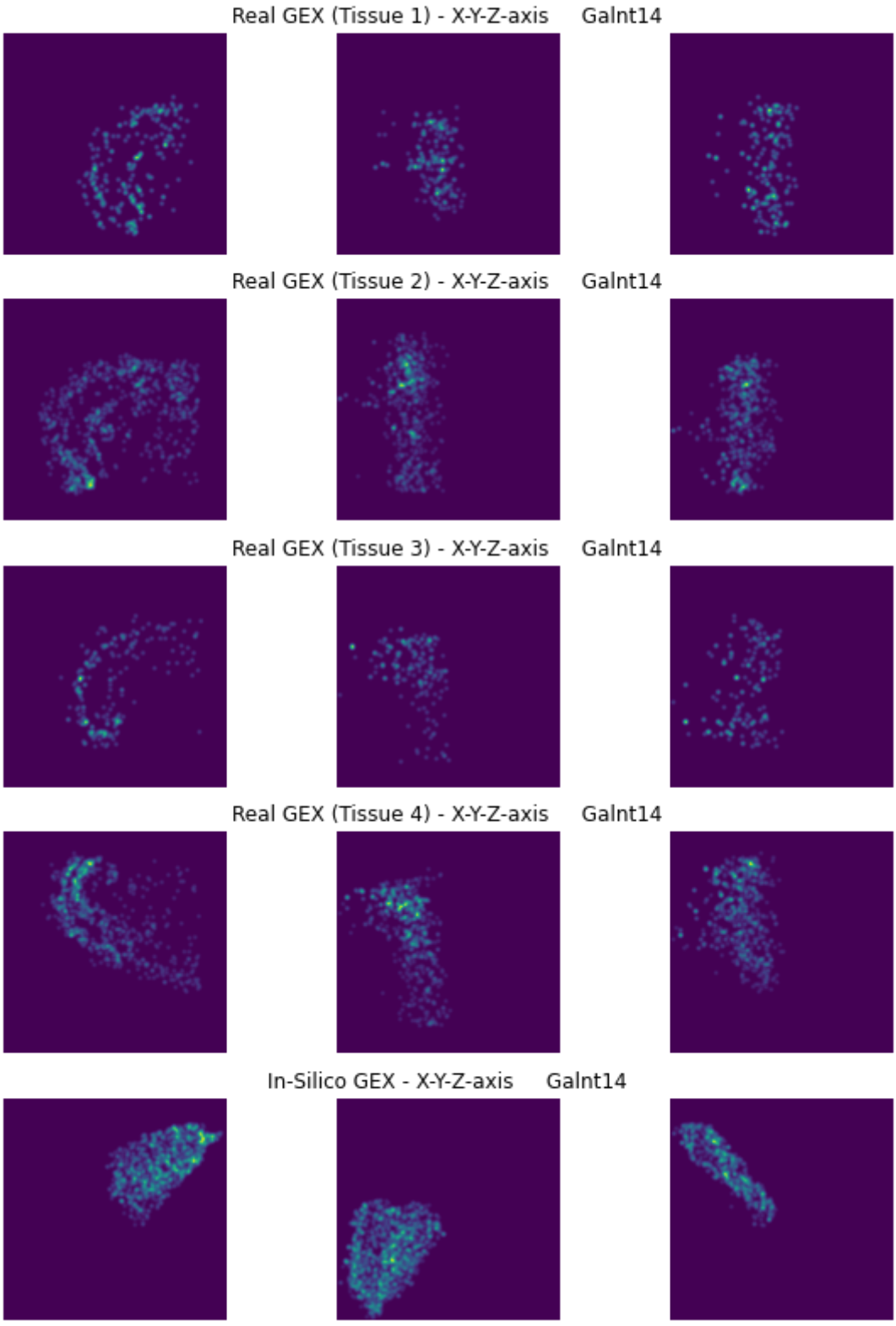
In-Silico GEX - X-Y-Z-axis Galntl6



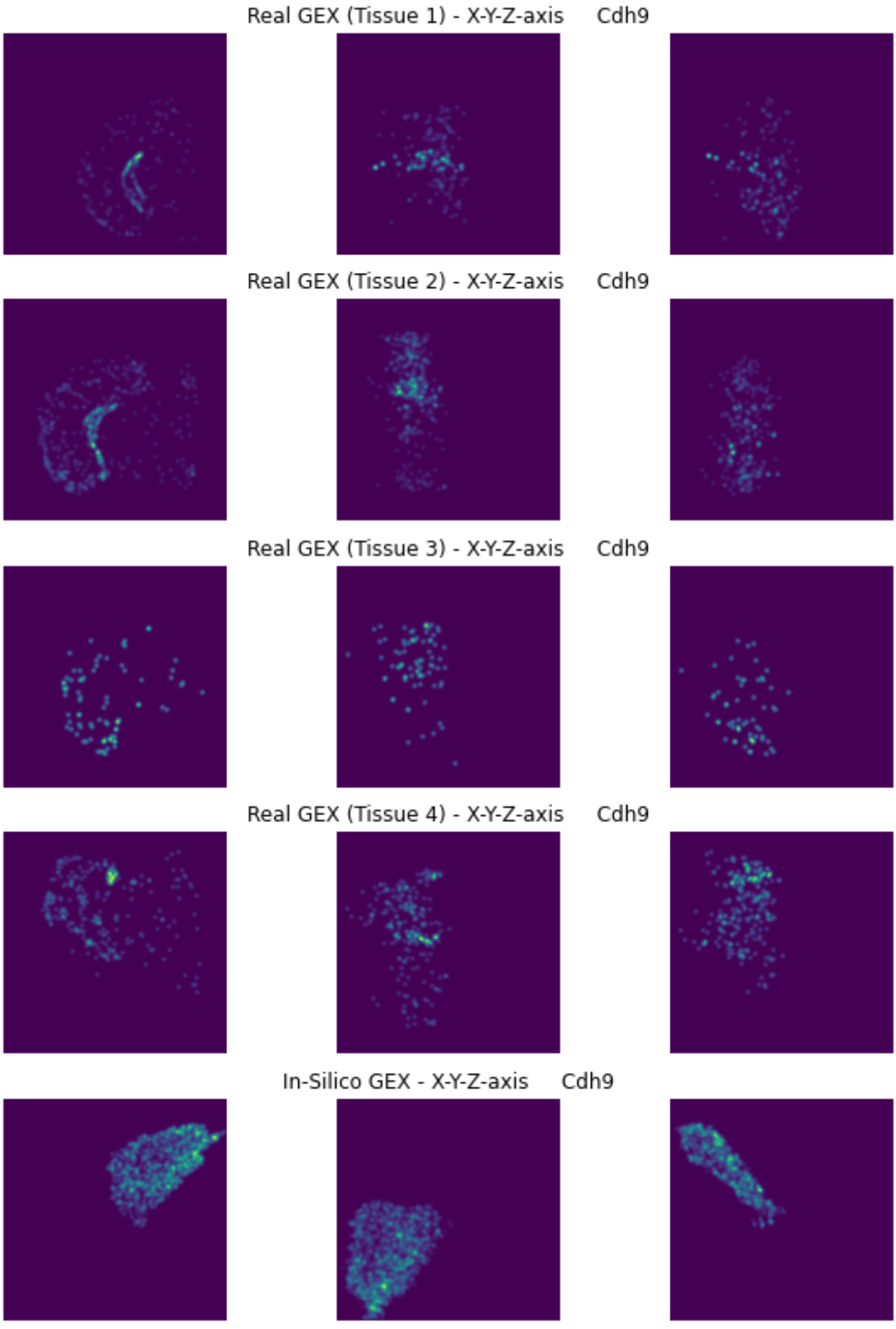
Real GEX Il1rapl2



Real GEX Galnt14

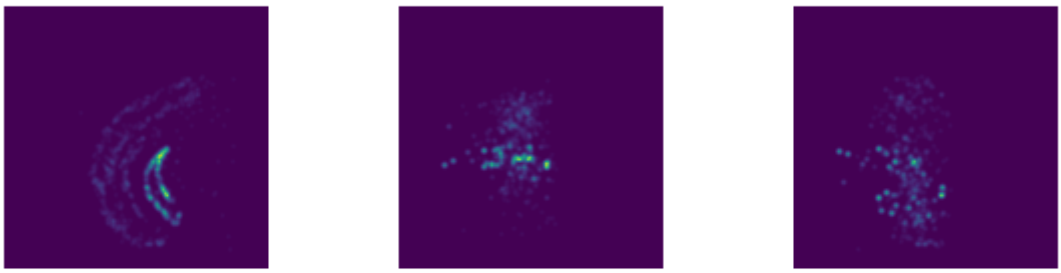


Real GEX Cdh9

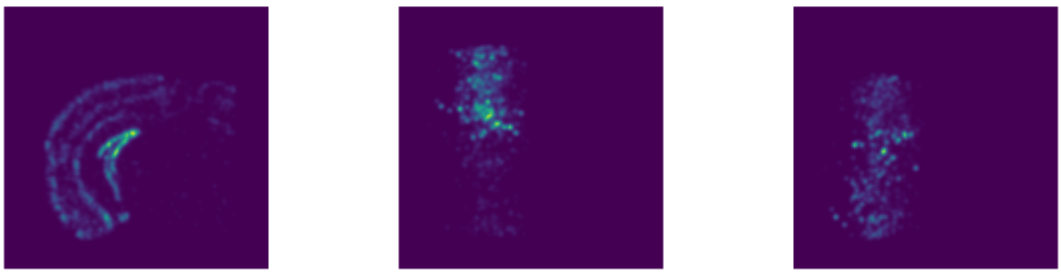


Real GEX Rasl10a

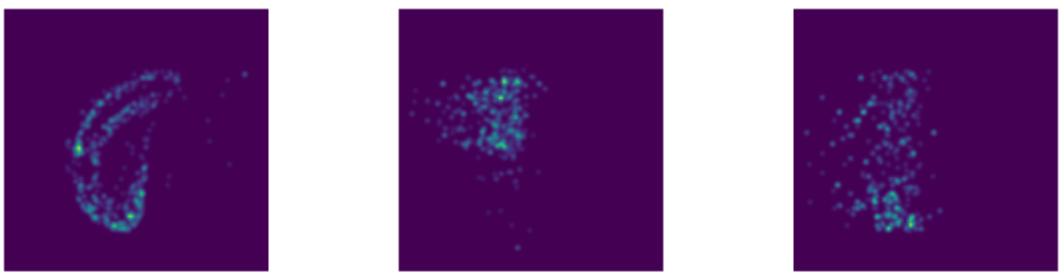
Real GEX (Tissue 1) - X-Y-Z-axis Rasl10a



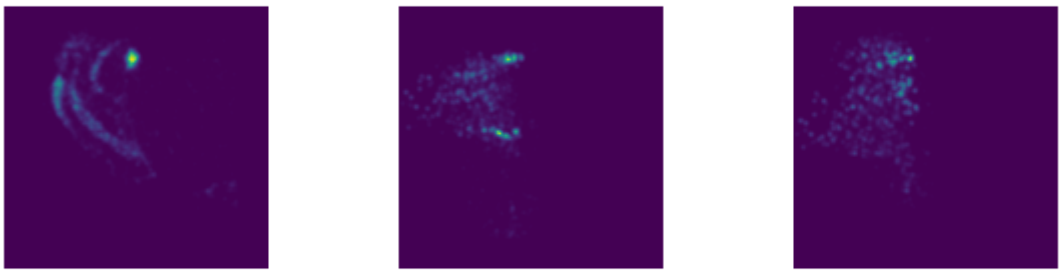
Real GEX (Tissue 2) - X-Y-Z-axis Rasl10a



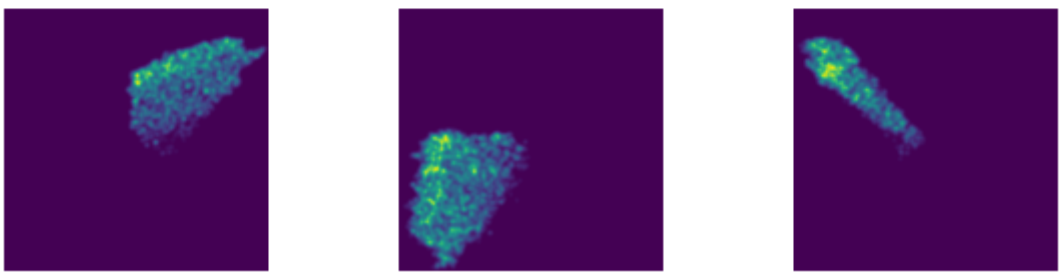
Real GEX (Tissue 3) - X-Y-Z-axis Rasl10a



Real GEX (Tissue 4) - X-Y-Z-axis Rasl10a

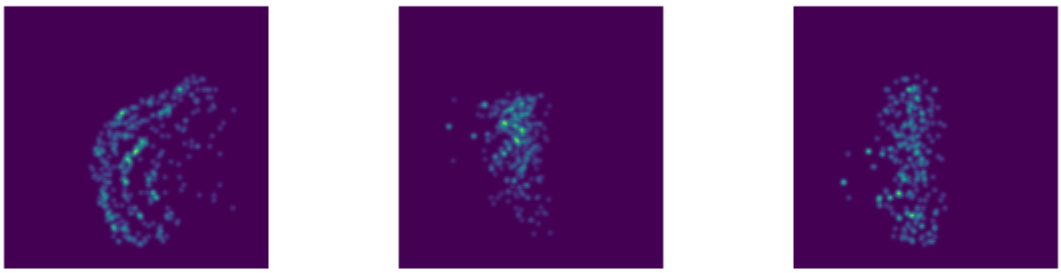


In-Silico GEX - X-Y-Z-axis Rasl10a

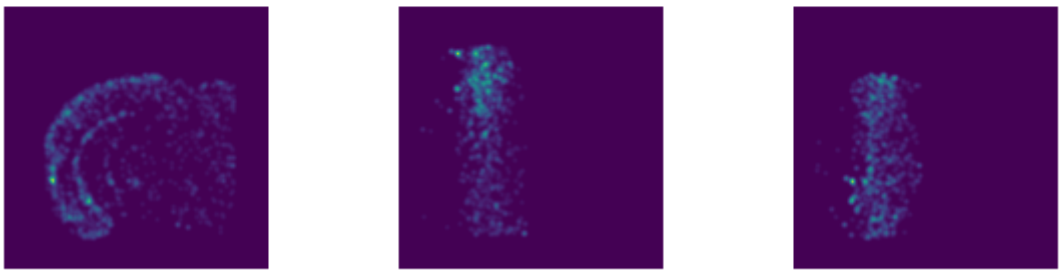


Real GEX Sorcs3

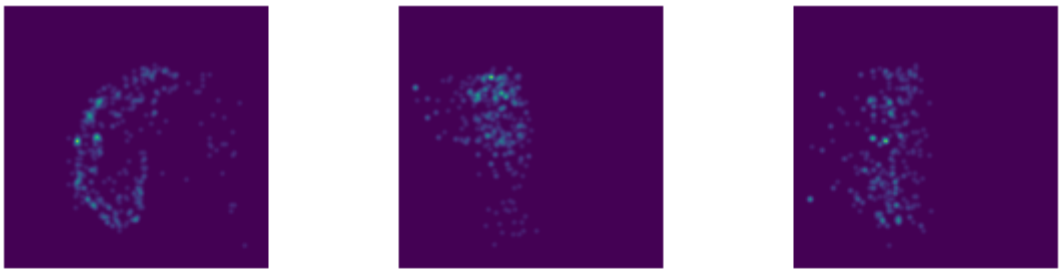
Real GEX (Tissue 1) - X-Y-Z-axis Sorcs3



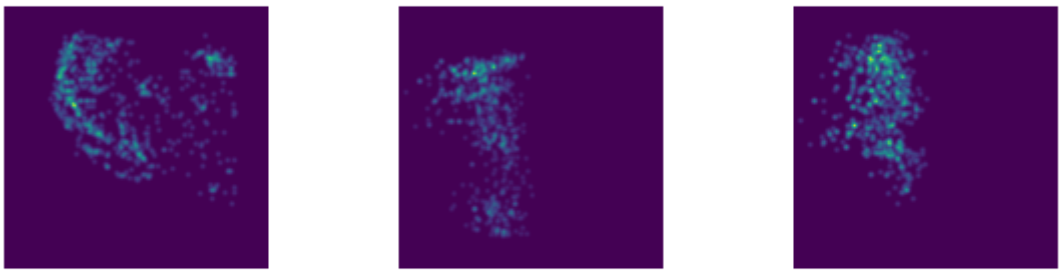
Real GEX (Tissue 2) - X-Y-Z-axis Sorcs3



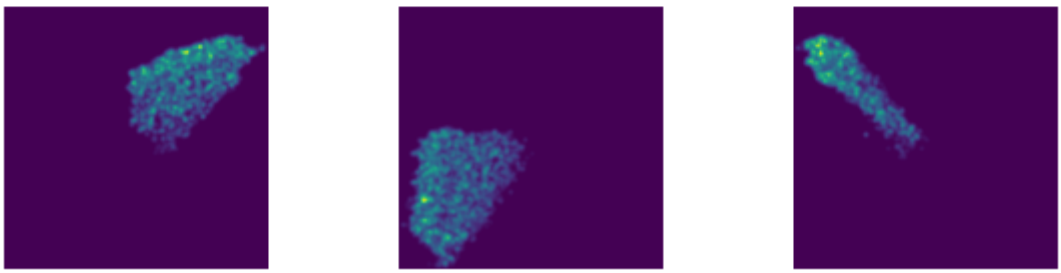
Real GEX (Tissue 3) - X-Y-Z-axis Sorcs3



Real GEX (Tissue 4) - X-Y-Z-axis Sorcs3



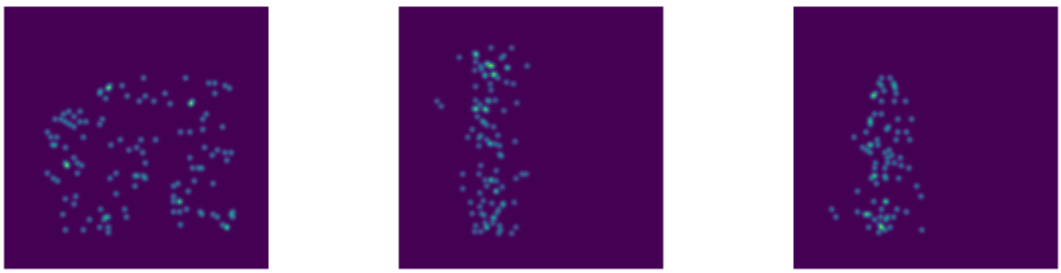
In-Silico GEX - X-Y-Z-axis Sorcs3



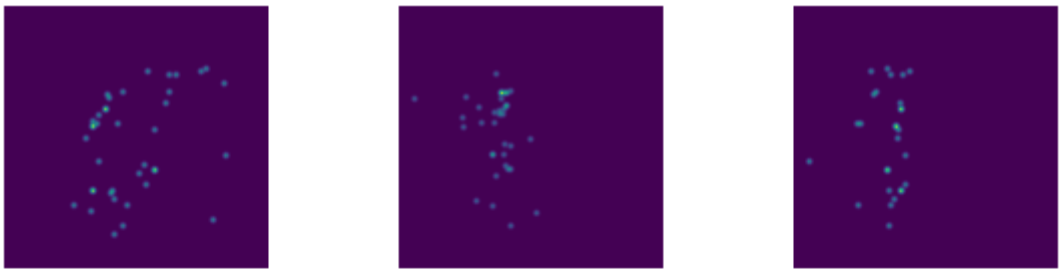
Real GEX Sdk1

Real GEX (Tissue 1) - X-Y-Z-axis Sdk1

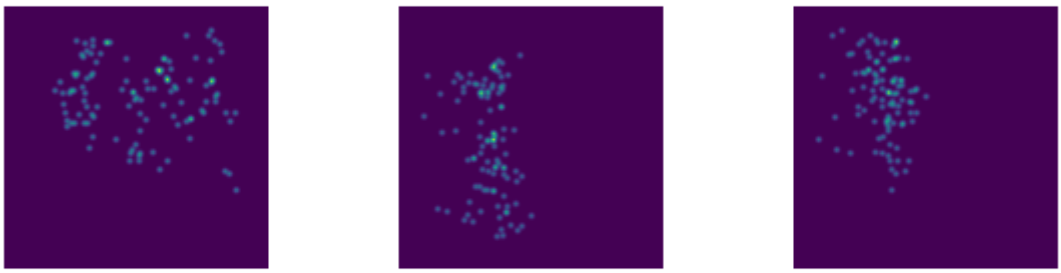
Real GEX (Tissue 2) - X-Y-Z-axis Sdk1



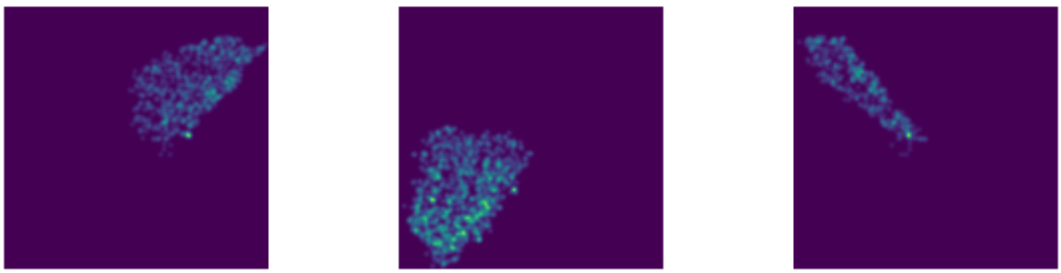
Real GEX (Tissue 3) - X-Y-Z-axis Sdk1



Real GEX (Tissue 4) - X-Y-Z-axis Sdk1

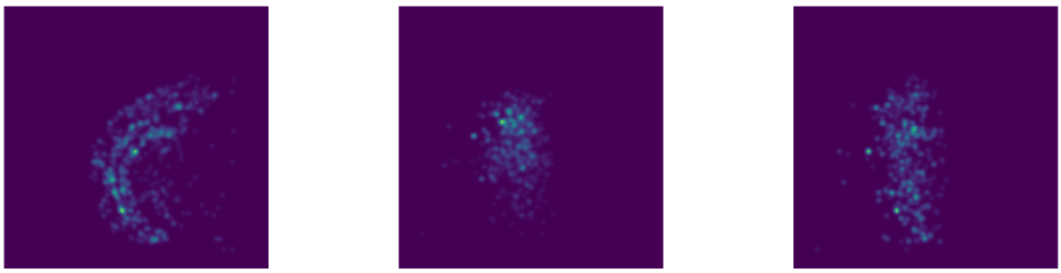


In-Silico GEX - X-Y-Z-axis Sdk1

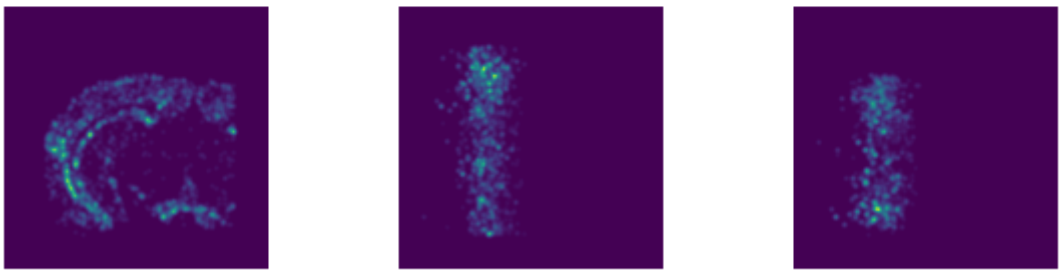


Real GEX Nr4a2

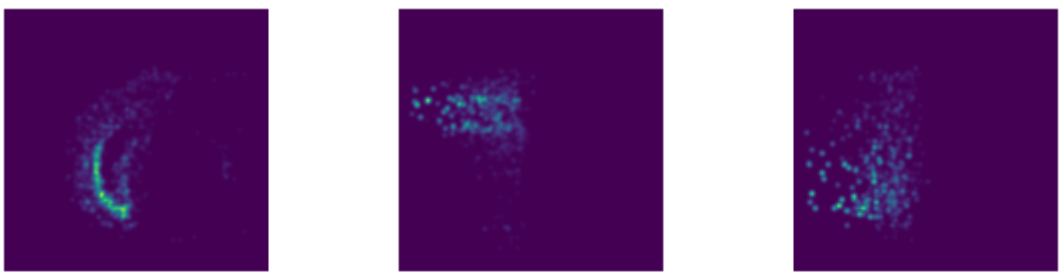
Real GEX (Tissue 1) - X-Y-Z-axis Nr4a2



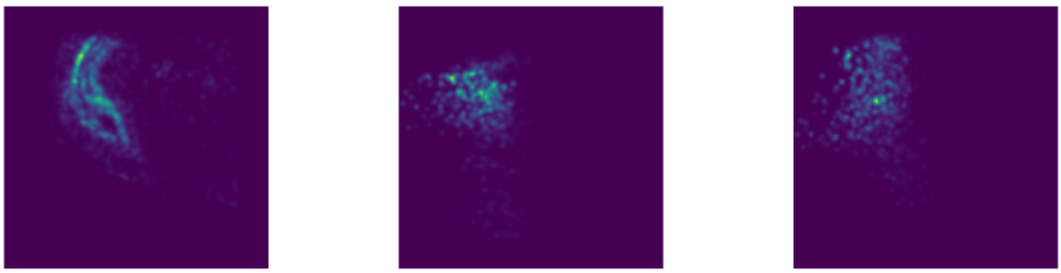
Real GEX (Tissue 2) - X-Y-Z-axis Nr4a2



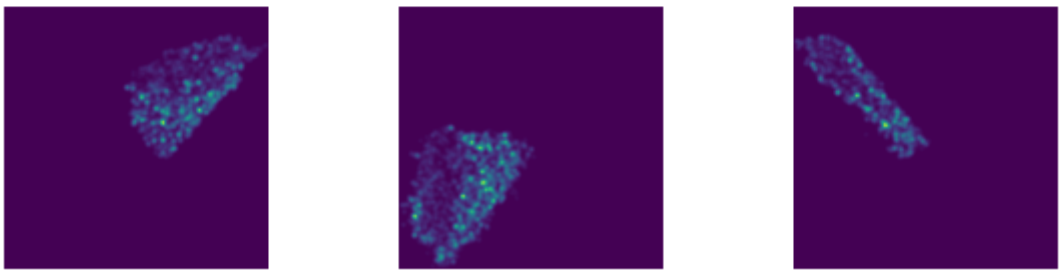
Real GEX (Tissue 3) - X-Y-Z-axis Nr4a2



Real GEX (Tissue 4) - X-Y-Z-axis Nr4a2

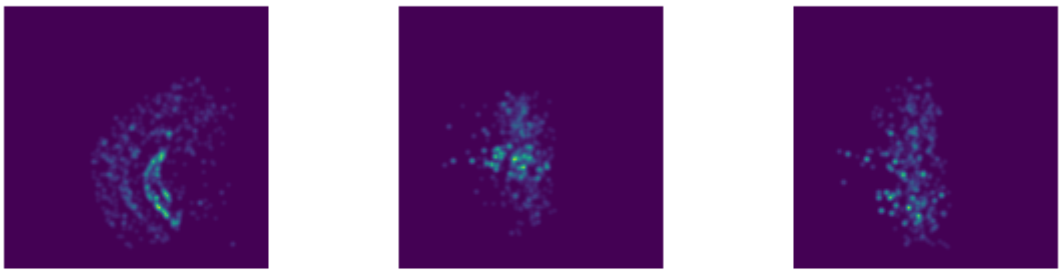


In-Silico GEX - X-Y-Z-axis Nr4a2

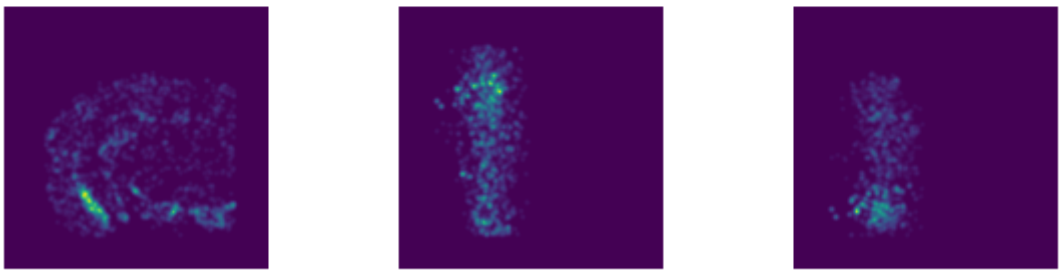


Real GEX Gfra1

Real GEX (Tissue 1) - X-Y-Z-axis Gfra1

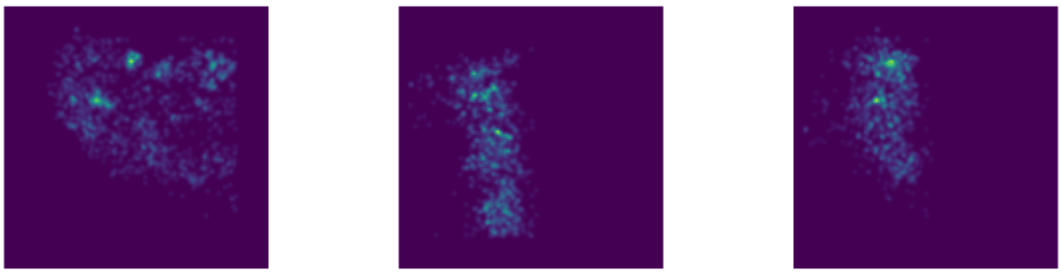


Real GEX (Tissue 2) - X-Y-Z-axis Gfra1



Real GEX (Tissue 3) - X-Y-Z-axis Gfra1

Real GEX (Tissue 4) - X-Y-Z-axis Gfra1

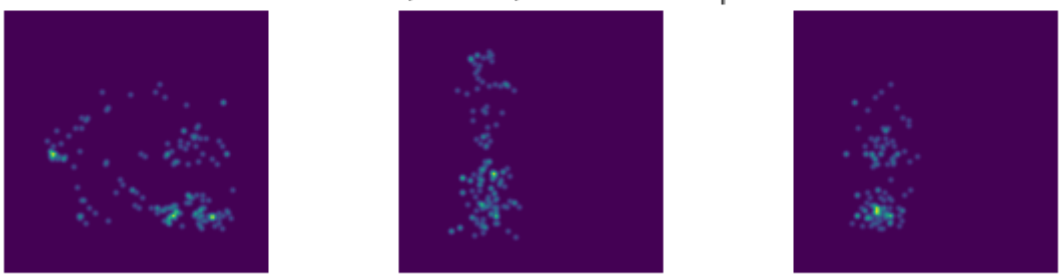


In-Silico GEX - X-Y-Z-axis Gfra1

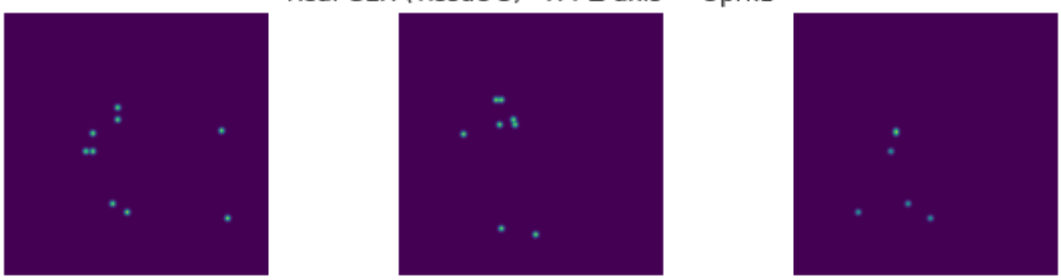
Real GEX Oprk1

Real GEX (Tissue 1) - X-Y-Z-axis Oprk1

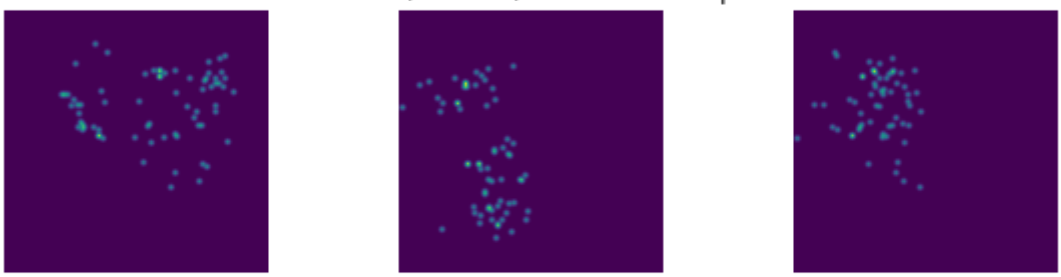
Real GEX (Tissue 2) - X-Y-Z-axis Oprk1



Real GEX (Tissue 3) - X-Y-Z-axis Oprk1



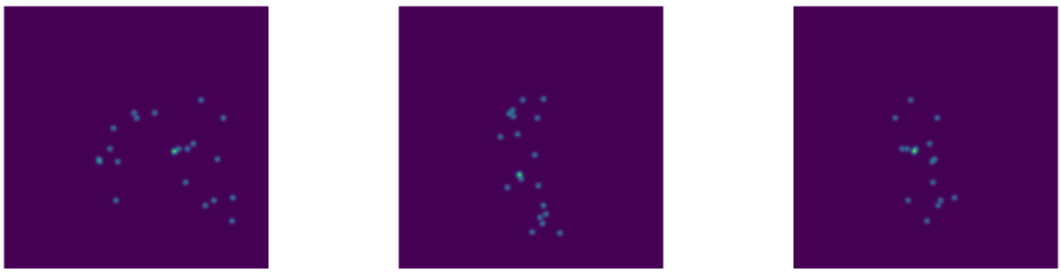
Real GEX (Tissue 4) - X-Y-Z-axis Oprk1



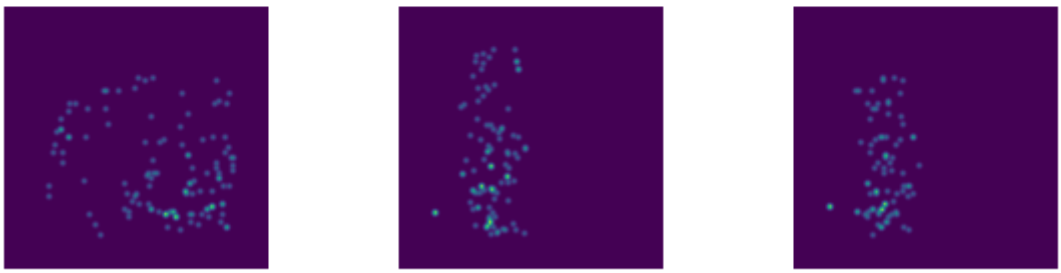
In-Silico GEX - X-Y-Z-axis Oprk1

Real GEX Col11a1

Real GEX (Tissue 1) - X-Y-Z-axis Col11a1

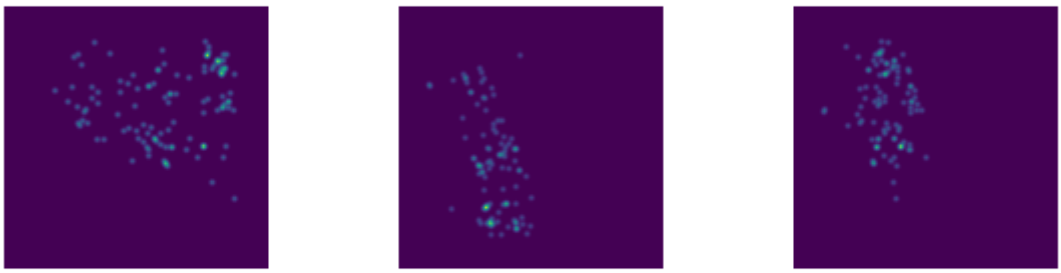


Real GEX (Tissue 2) - X-Y-Z-axis Col11a1

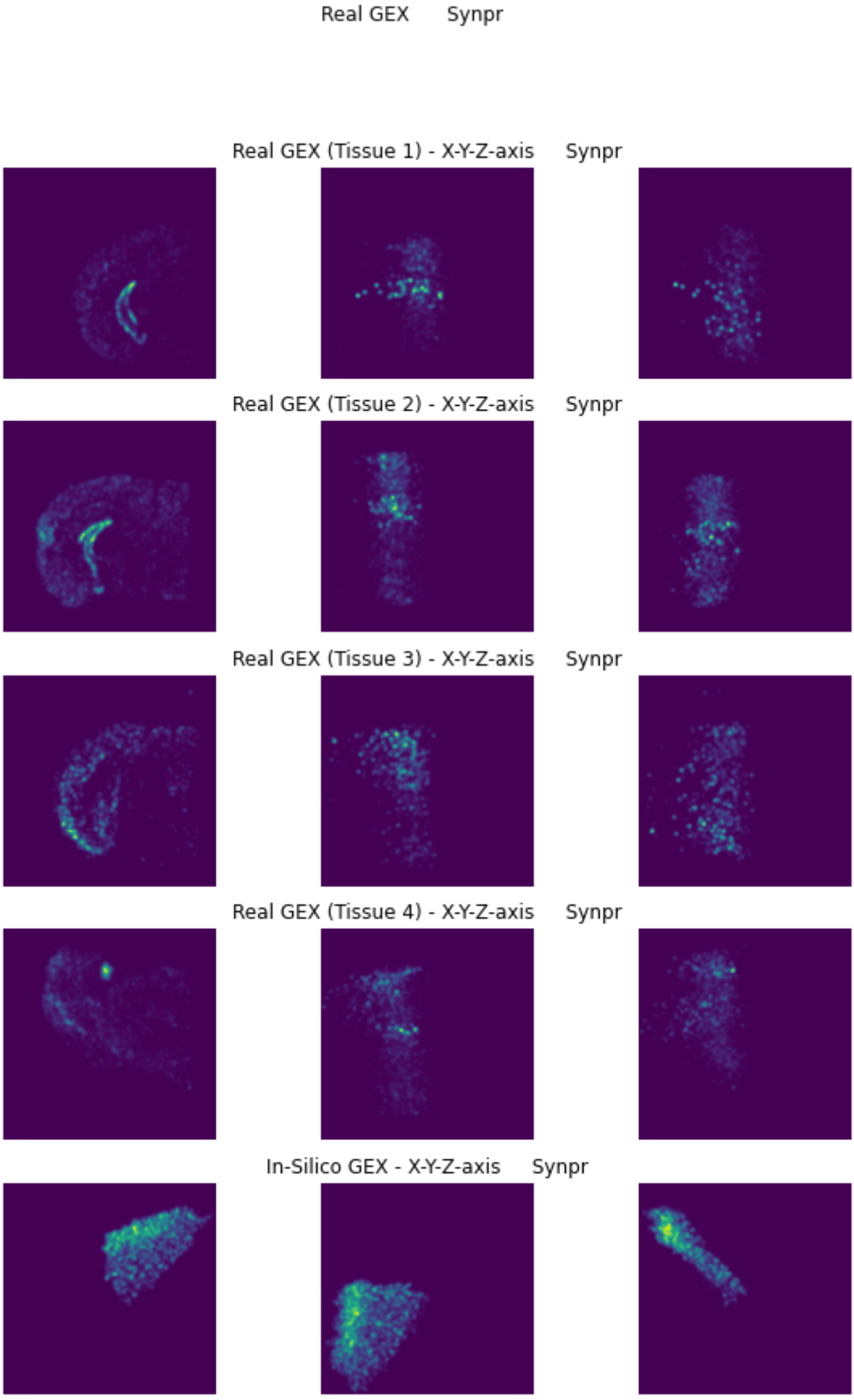


Real GEX (Tissue 3) - X-Y-Z-axis Col11a1

Real GEX (Tissue 4) - X-Y-Z-axis Col11a1

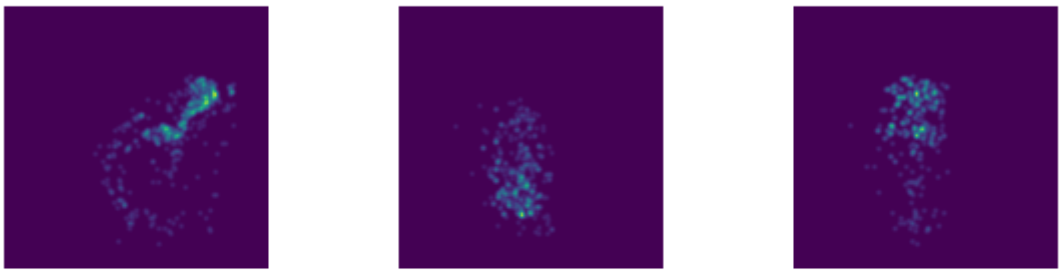


In-Silico GEX - X-Y-Z-axis Col11a1

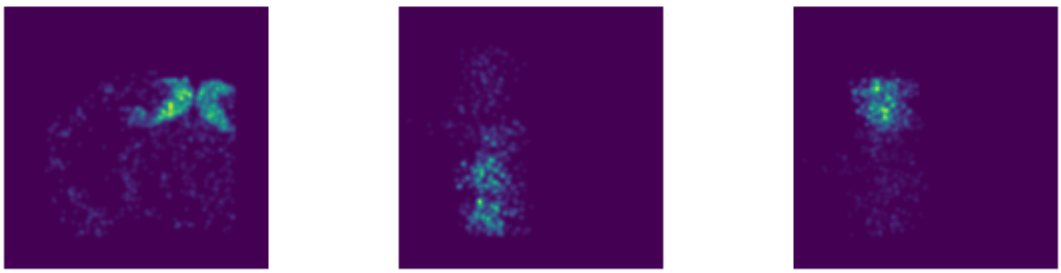

Real GEX
Synpr
Real GEX (Tissue 1) - X-Y-Z-axis
Synpr
Real GEX (Tissue 2) - X-Y-Z-axis
Synpr
Real GEX (Tissue 3) - X-Y-Z-axis
Synpr
Real GEX (Tissue 4) - X-Y-Z-axis
Synpr
In-Silico GEX - X-Y-Z-axis
Synpr

Real GEX Tshz2

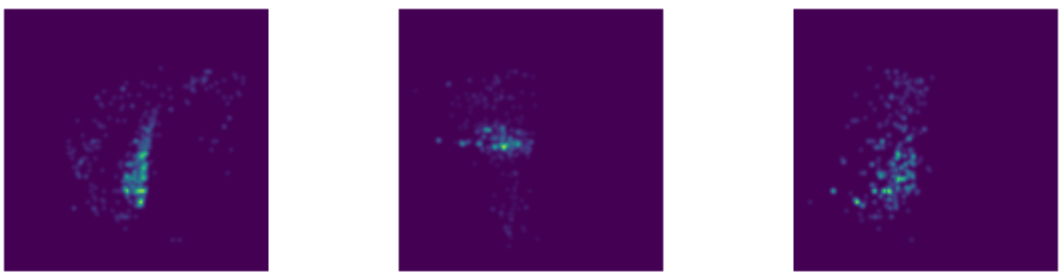
Real GEX (Tissue 1) - X-Y-Z-axis Tshz2



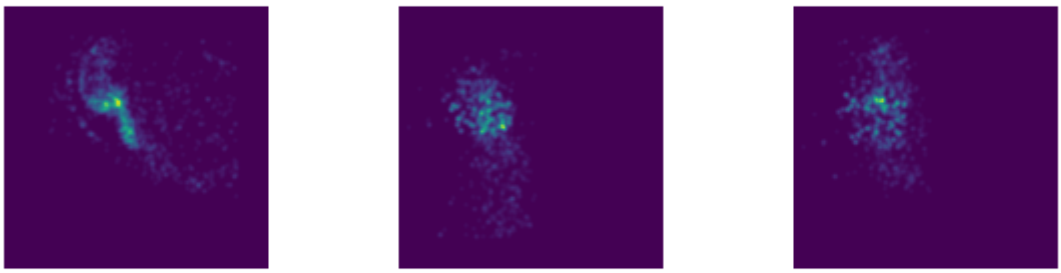
Real GEX (Tissue 2) - X-Y-Z-axis Tshz2



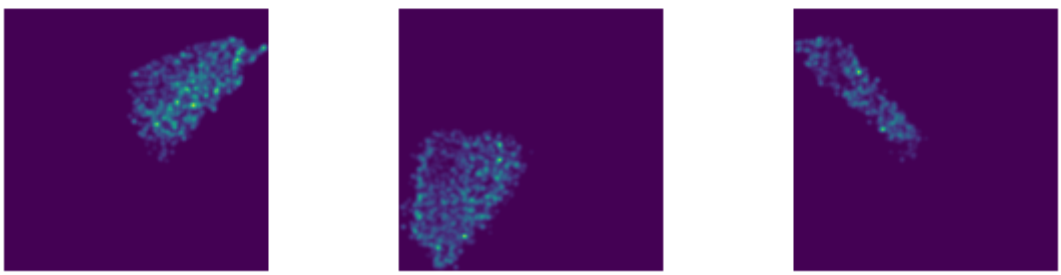
Real GEX (Tissue 3) - X-Y-Z-axis Tshz2



Real GEX (Tissue 4) - X-Y-Z-axis Tshz2

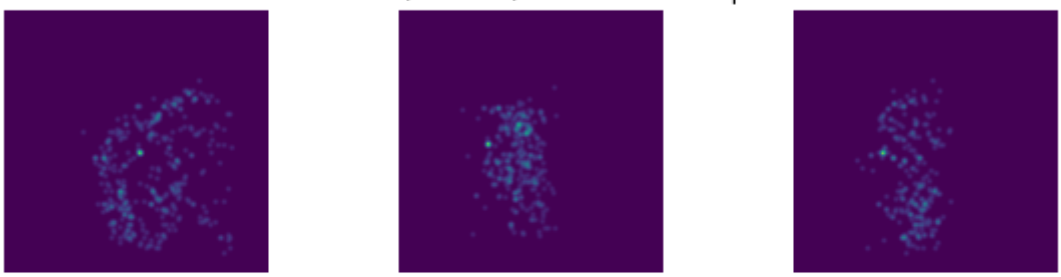


In-Silico GEX - X-Y-Z-axis Tshz2

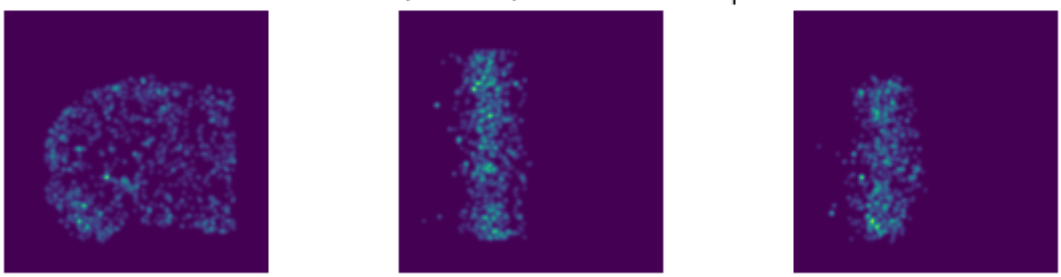


Real GEX Nxph1

Real GEX (Tissue 1) - X-Y-Z-axis Nxph1

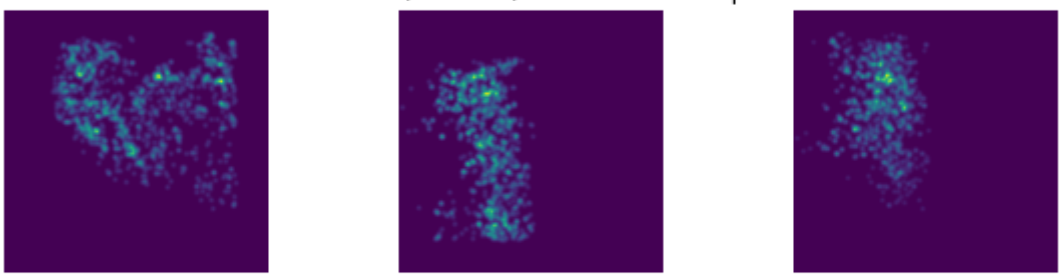


Real GEX (Tissue 2) - X-Y-Z-axis Nxph1

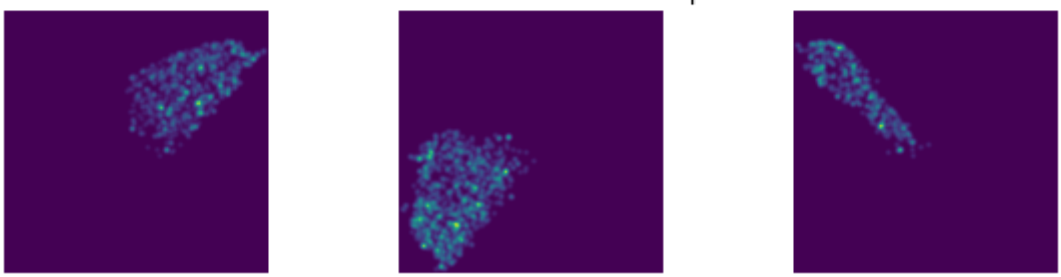


Real GEX (Tissue 3) - X-Y-Z-axis Nxph1

Real GEX (Tissue 4) - X-Y-Z-axis Nxph1

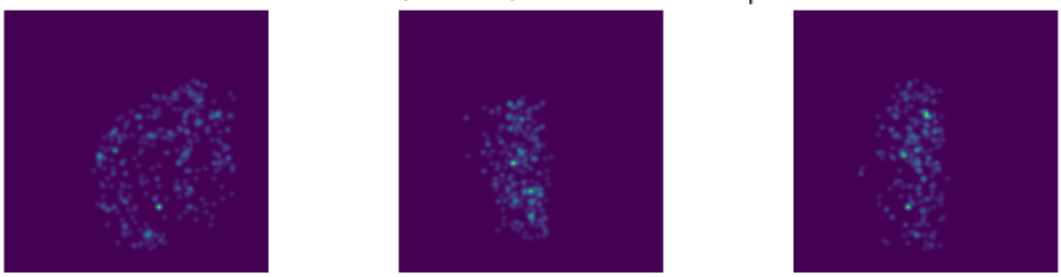


In-Silico GEX - X-Y-Z-axis Nxph1

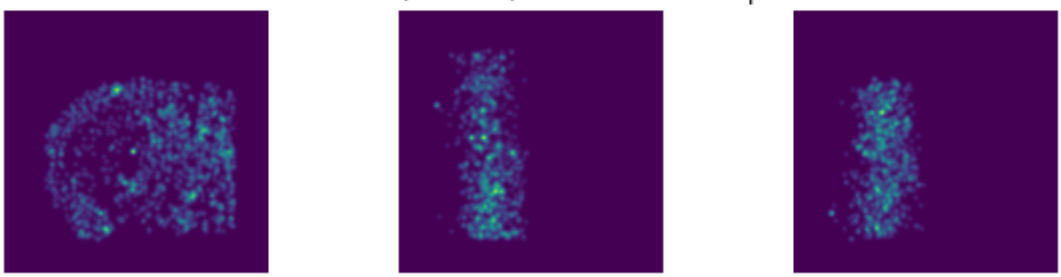


Real GEX Kcnip1

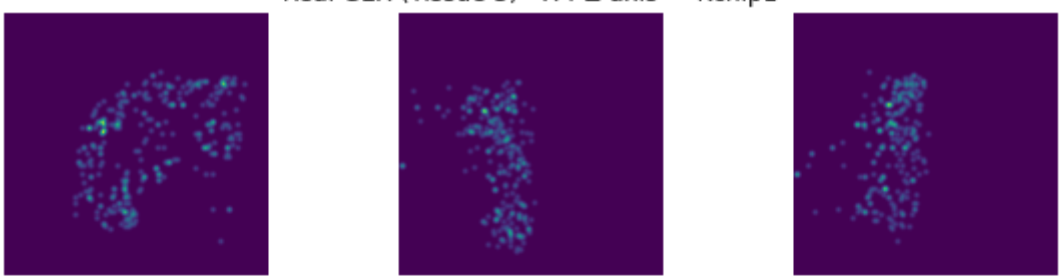
Real GEX (Tissue 1) - X-Y-Z-axis Kcnip1



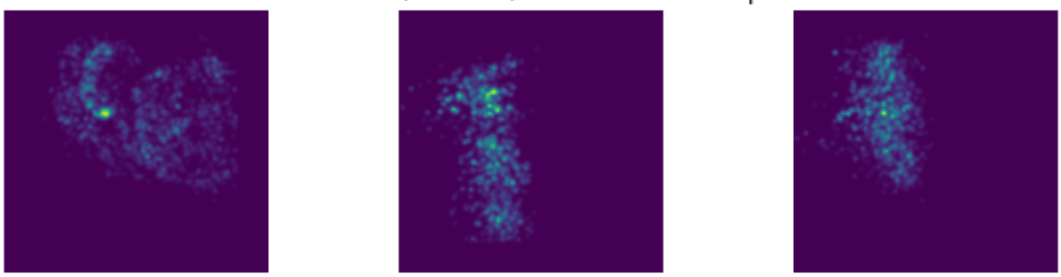
Real GEX (Tissue 2) - X-Y-Z-axis Kcnip1



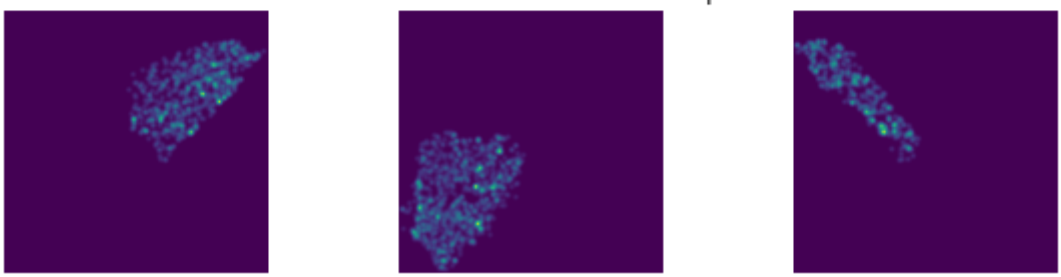
Real GEX (Tissue 3) - X-Y-Z-axis Kcnip1



Real GEX (Tissue 4) - X-Y-Z-axis Kcnip1

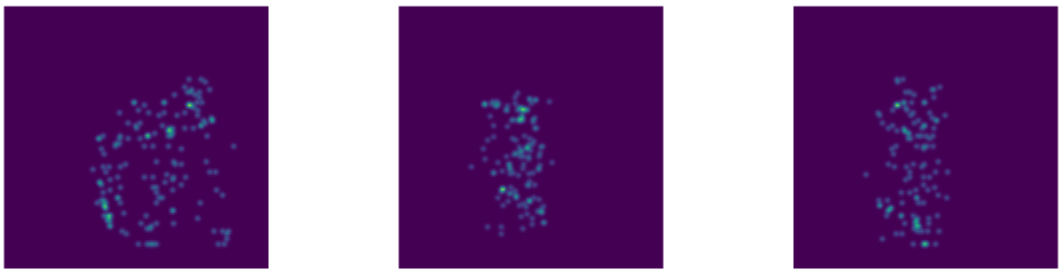


In-Silico GEX - X-Y-Z-axis Kcnip1

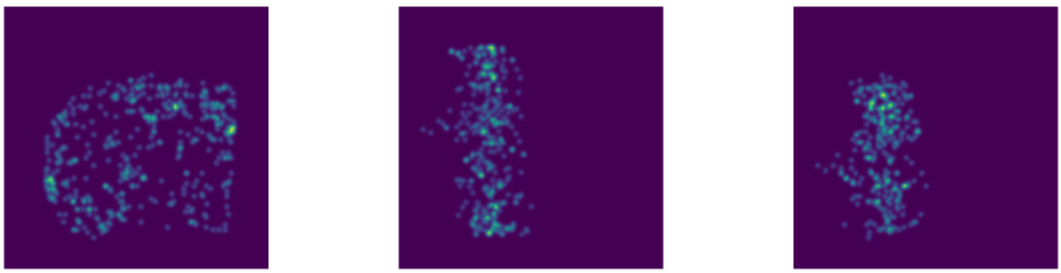


Real GEX Grik1

Real GEX (Tissue 1) - X-Y-Z-axis Grik1

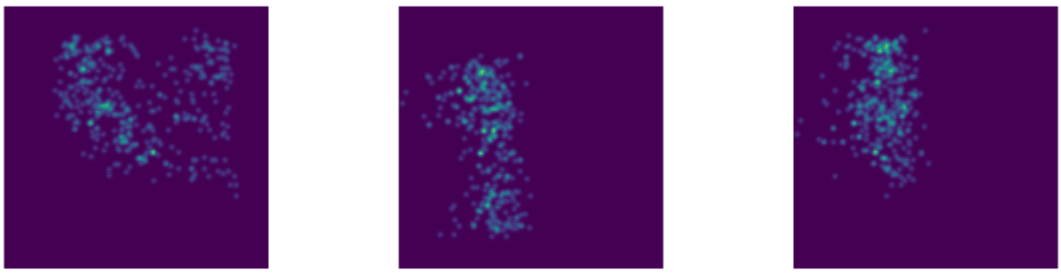


Real GEX (Tissue 2) - X-Y-Z-axis Grik1

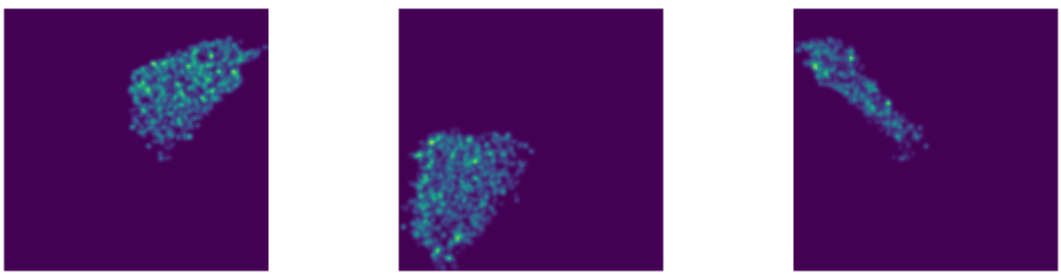


Real GEX (Tissue 3) - X-Y-Z-axis Grik1

Real GEX (Tissue 4) - X-Y-Z-axis Grik1

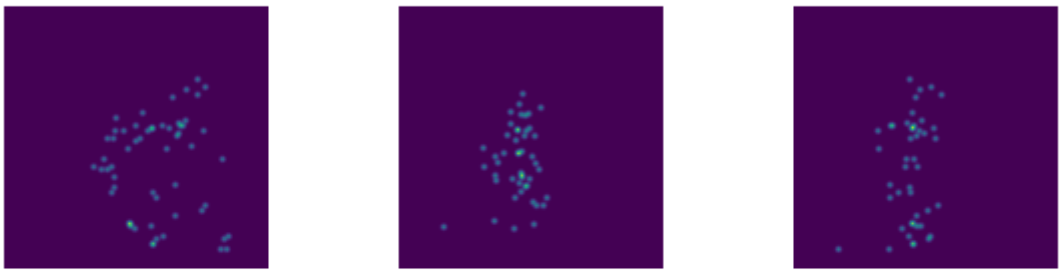


In-Silico GEX - X-Y-Z-axis Grik1

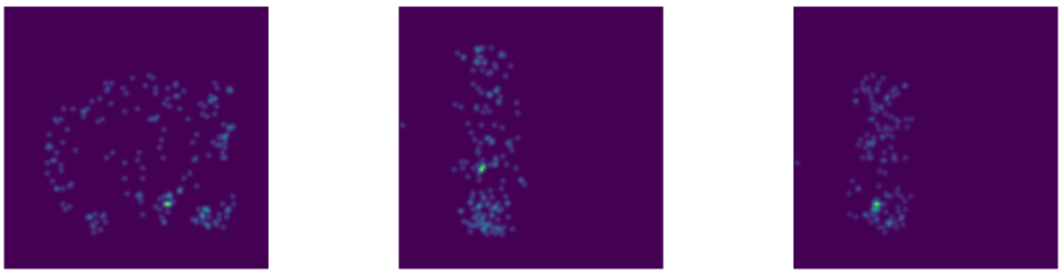


Real GEX Vwc2l

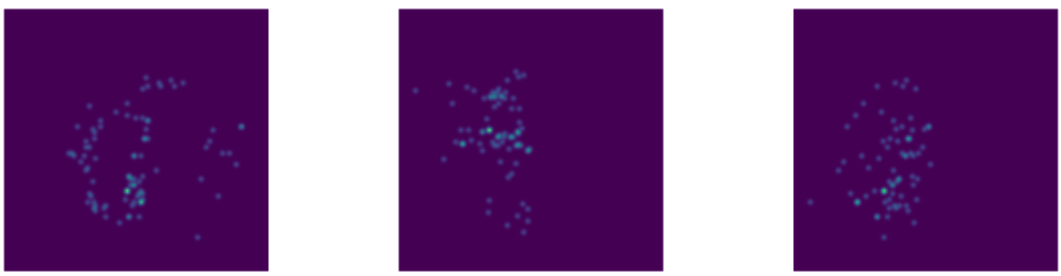
Real GEX (Tissue 1) - X-Y-Z-axis Vwc2l



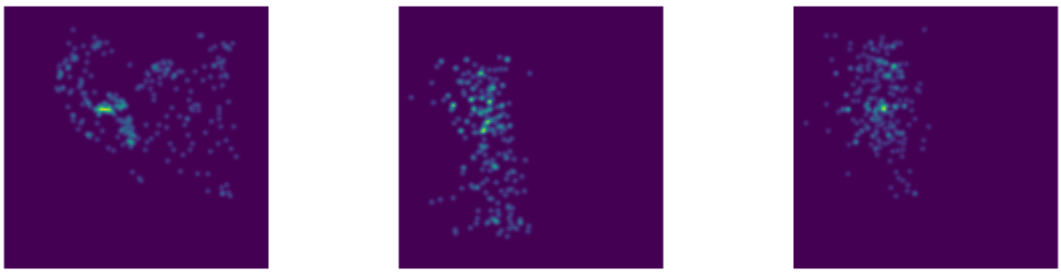
Real GEX (Tissue 2) - X-Y-Z-axis Vwc2l



Real GEX (Tissue 3) - X-Y-Z-axis Vwc2l



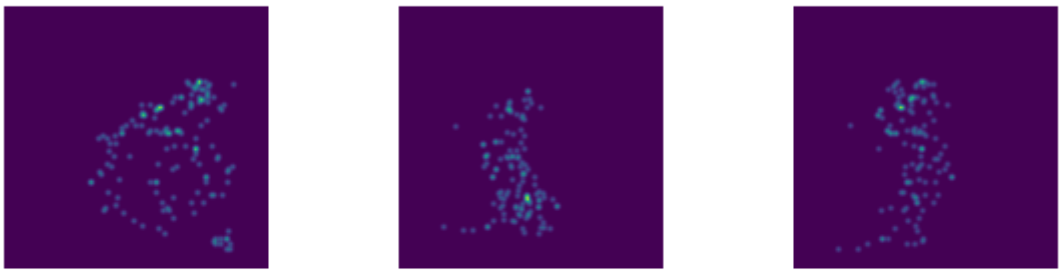
Real GEX (Tissue 4) - X-Y-Z-axis Vwc2l



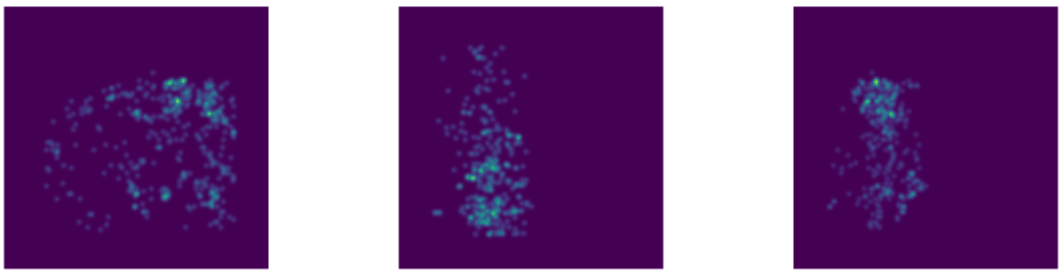
In-Silico GEX - X-Y-Z-axis Vwc2l

Real GEX Coro6

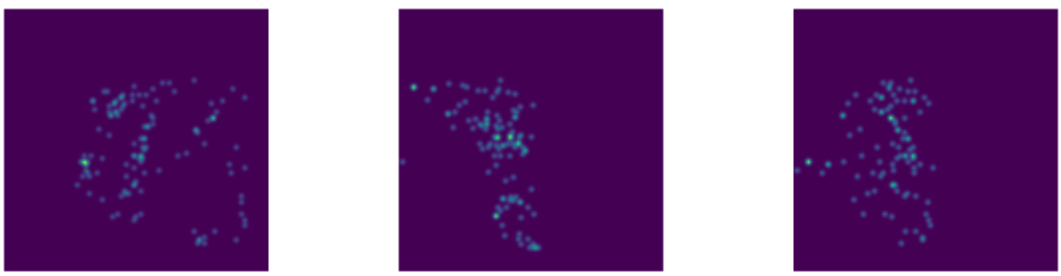
Real GEX (Tissue 1) - X-Y-Z-axis Coro6



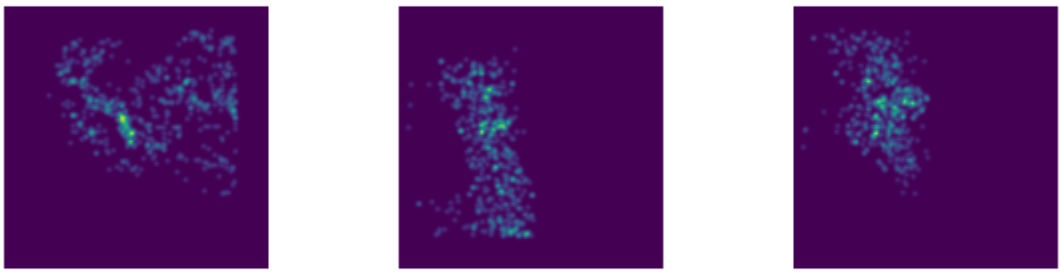
Real GEX (Tissue 2) - X-Y-Z-axis Coro6



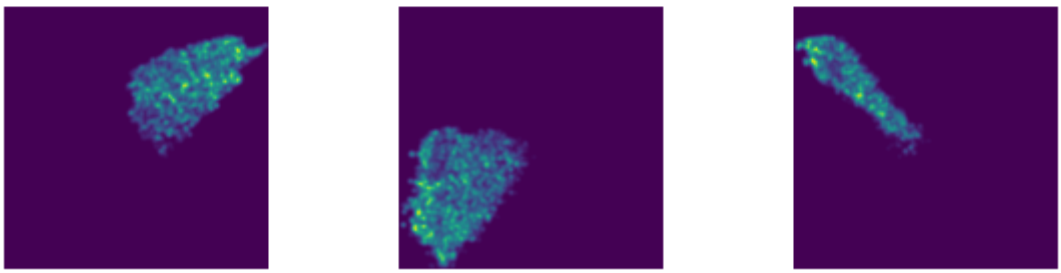
Real GEX (Tissue 3) - X-Y-Z-axis Coro6



Real GEX (Tissue 4) - X-Y-Z-axis Coro6

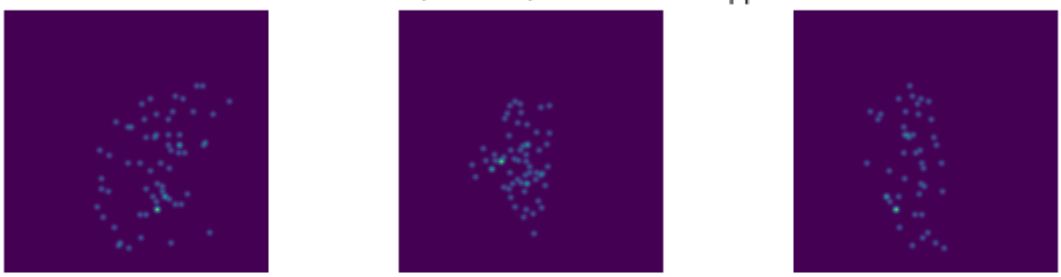


In-Silico GEX - X-Y-Z-axis Coro6

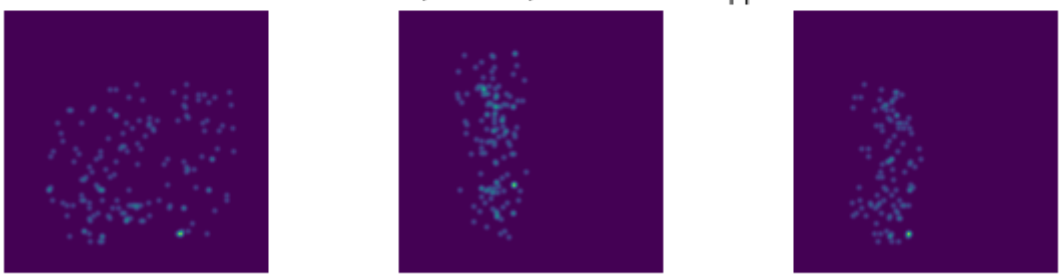


Real GEX Lpp

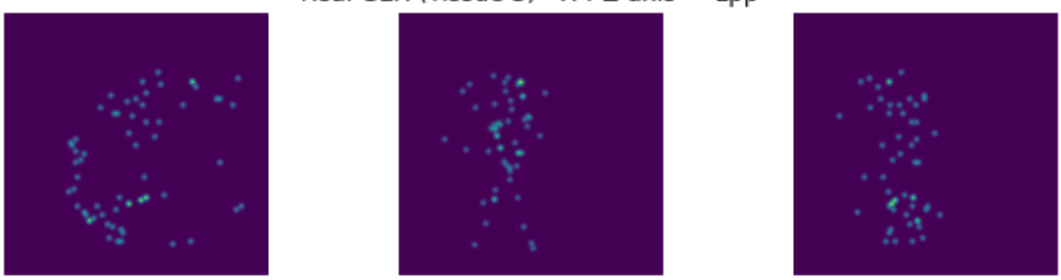
Real GEX (Tissue 1) - X-Y-Z-axis Lpp



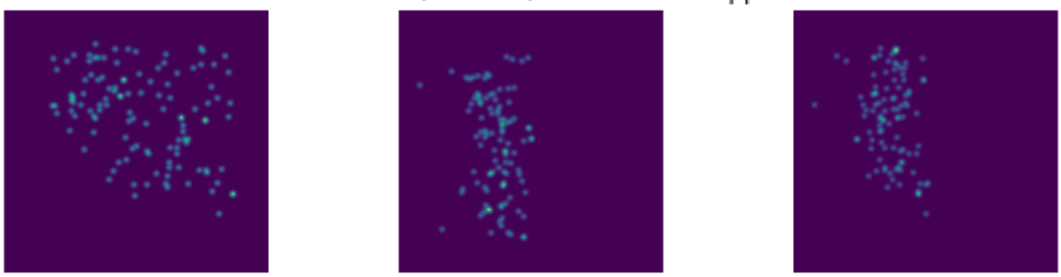
Real GEX (Tissue 2) - X-Y-Z-axis Lpp



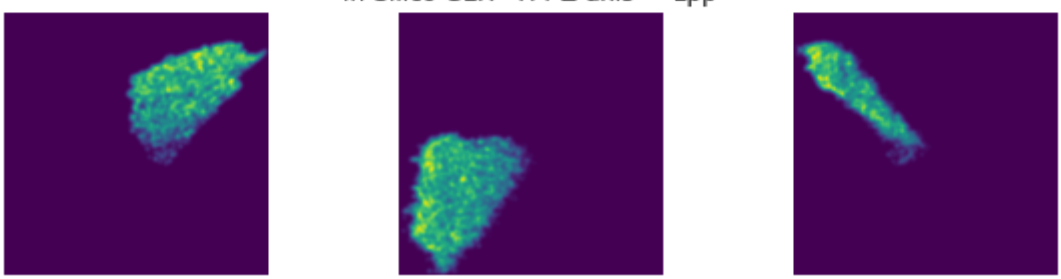
Real GEX (Tissue 3) - X-Y-Z-axis Lpp



Real GEX (Tissue 4) - X-Y-Z-axis Lpp

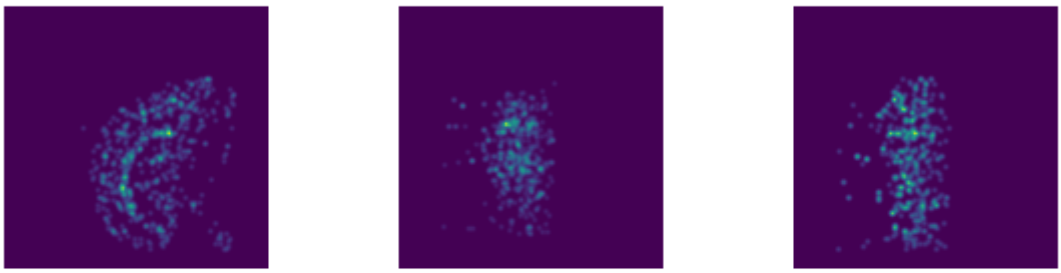


In-Silico GEX - X-Y-Z-axis Lpp

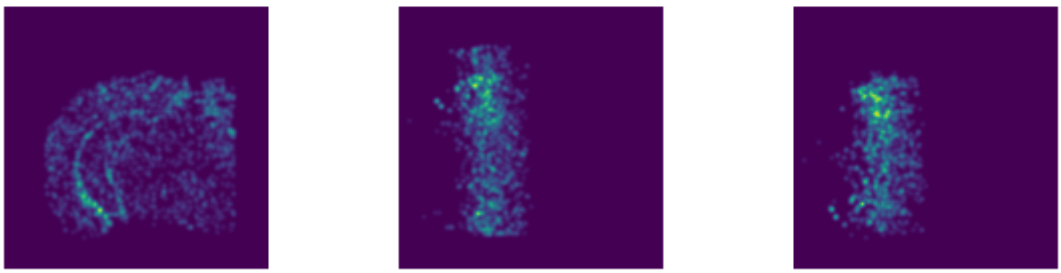


Real GEX Kcnn2

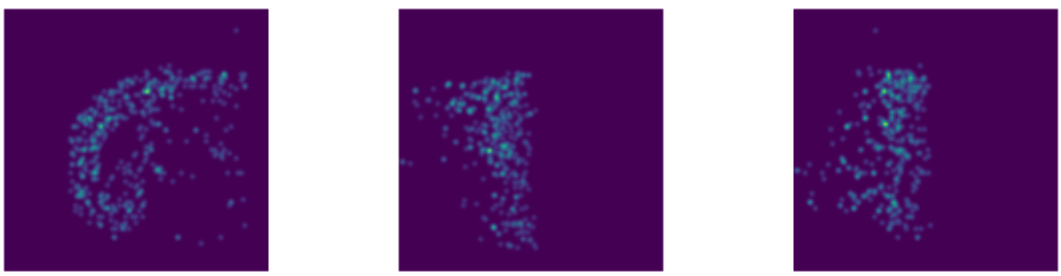
Real GEX (Tissue 1) - X-Y-Z-axis Kcnn2



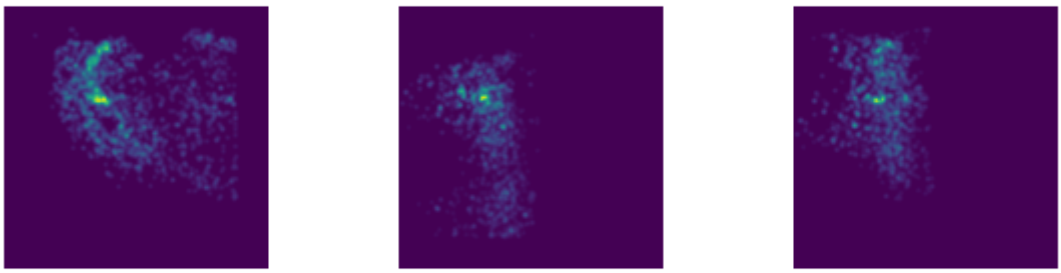
Real GEX (Tissue 2) - X-Y-Z-axis Kcnn2



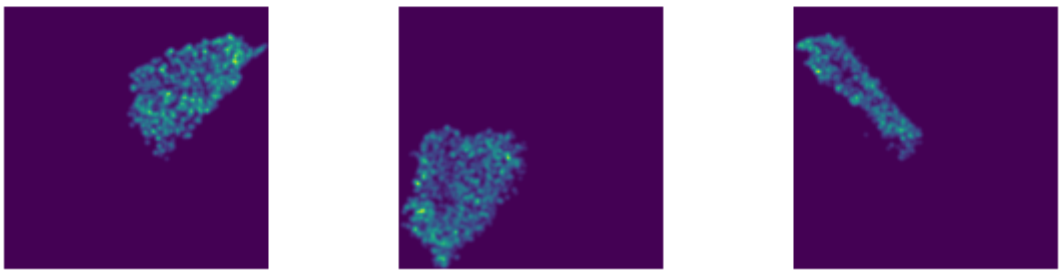
Real GEX (Tissue 3) - X-Y-Z-axis Kcnn2



Real GEX (Tissue 4) - X-Y-Z-axis Kcnn2

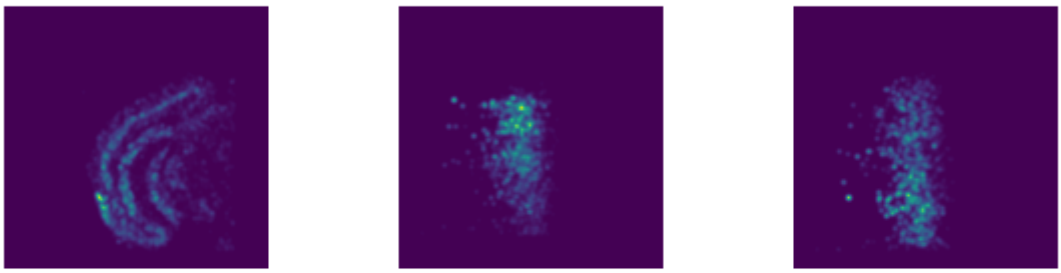


In-Silico GEX - X-Y-Z-axis Kcnn2

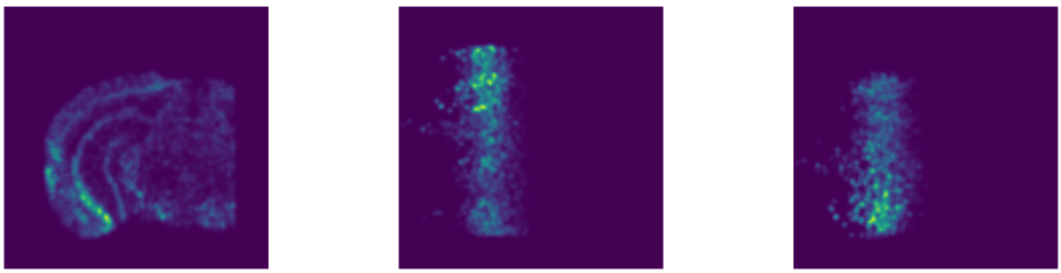


Real GEX Rab3c

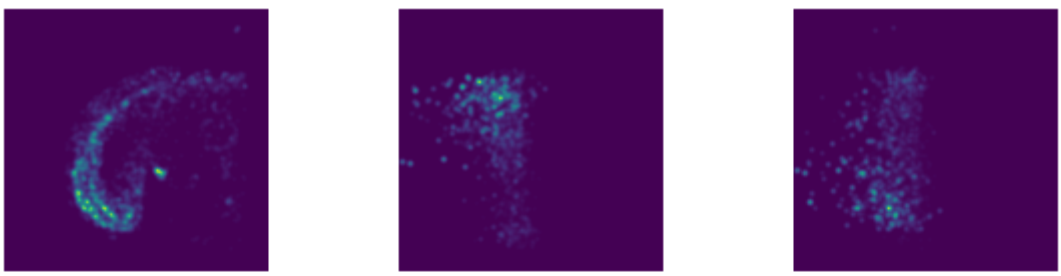
Real GEX (Tissue 1) - X-Y-Z-axis Rab3c



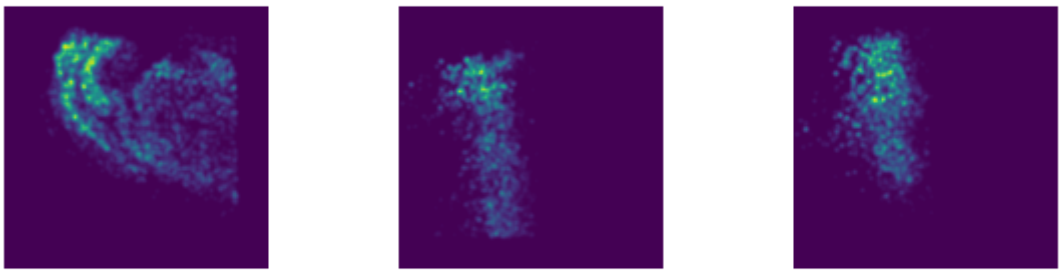
Real GEX (Tissue 2) - X-Y-Z-axis Rab3c



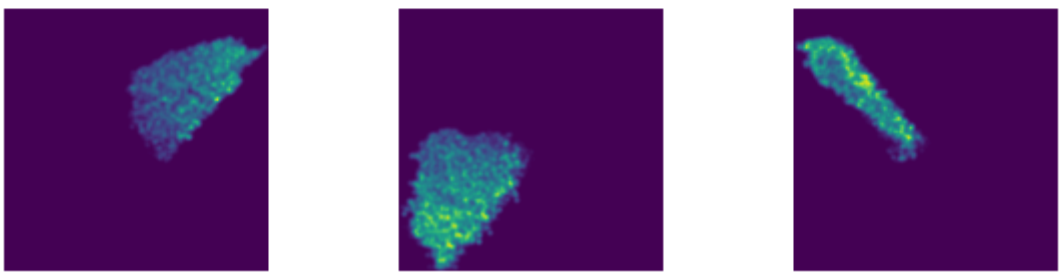
Real GEX (Tissue 3) - X-Y-Z-axis Rab3c



Real GEX (Tissue 4) - X-Y-Z-axis Rab3c

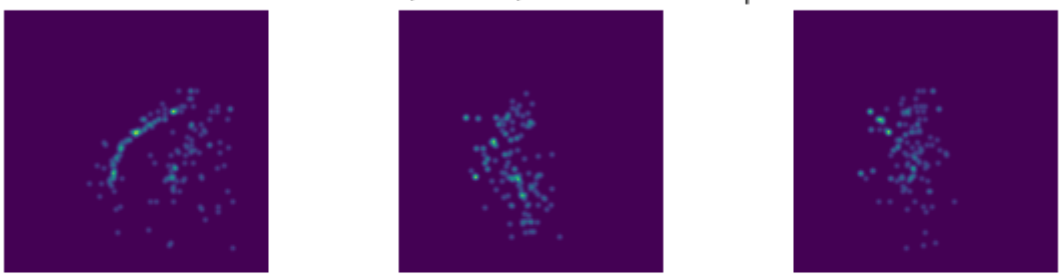


In-Silico GEX - X-Y-Z-axis Rab3c

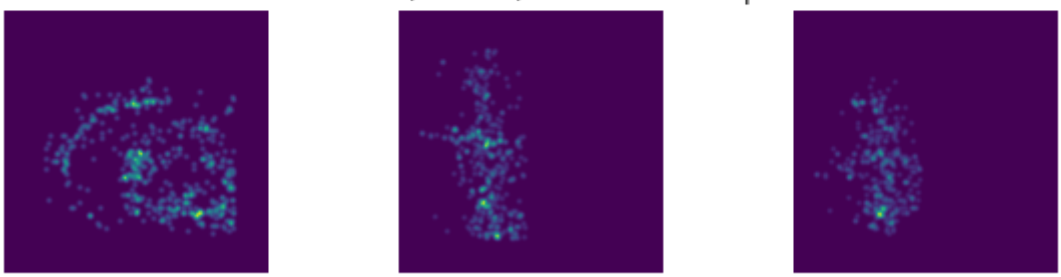


Real GEX Foxp2

Real GEX (Tissue 1) - X-Y-Z-axis Foxp2

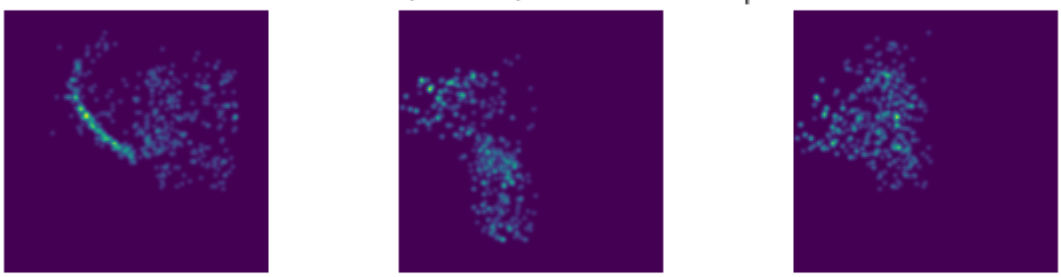


Real GEX (Tissue 2) - X-Y-Z-axis Foxp2

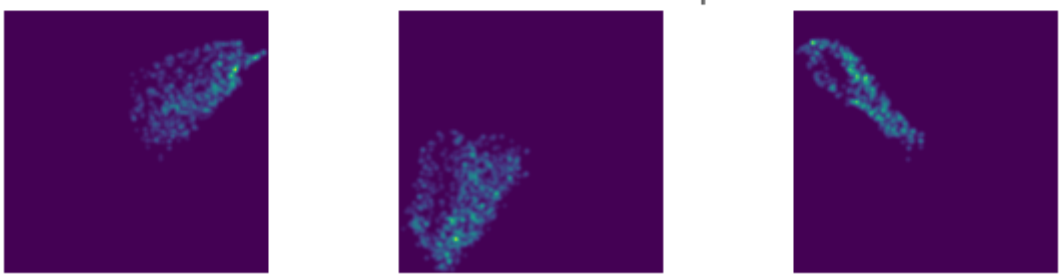


Real GEX (Tissue 3) - X-Y-Z-axis Foxp2

Real GEX (Tissue 4) - X-Y-Z-axis Foxp2

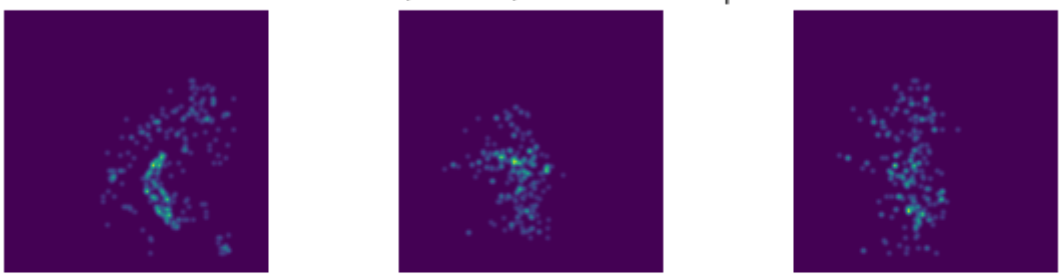


In-Silico GEX - X-Y-Z-axis Foxp2

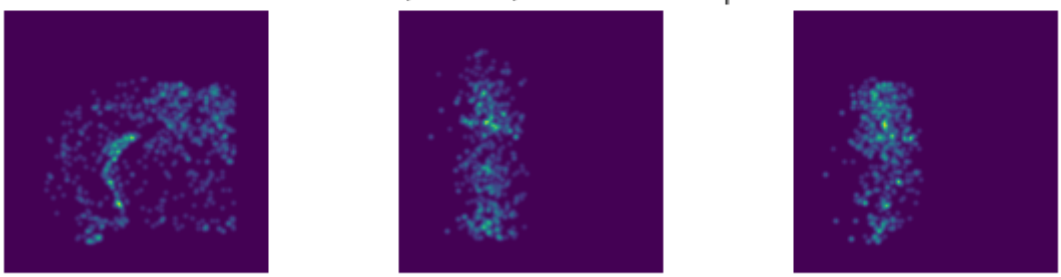


Real GEX Zfpm2

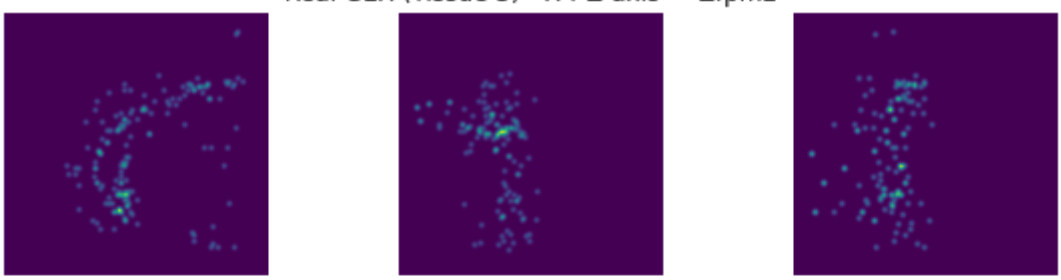
Real GEX (Tissue 1) - X-Y-Z-axis Zfpm2



Real GEX (Tissue 2) - X-Y-Z-axis Zfpm2

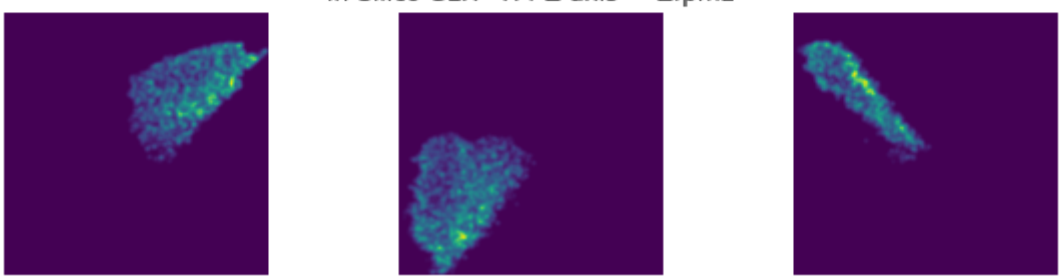


Real GEX (Tissue 3) - X-Y-Z-axis Zfpm2



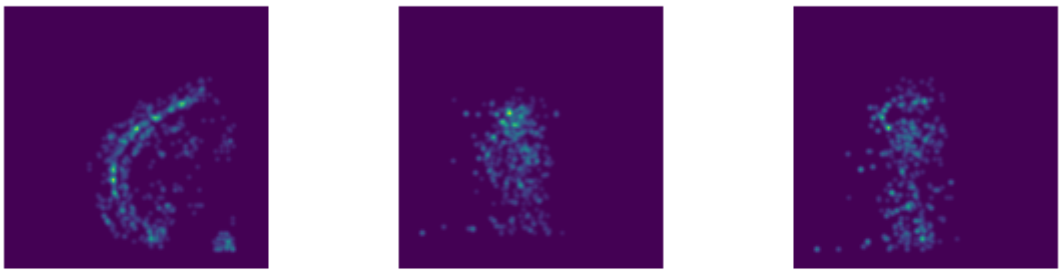
Real GEX (Tissue 4) - X-Y-Z-axis Zfpm2

In-Silico GEX - X-Y-Z-axis Zfpm2

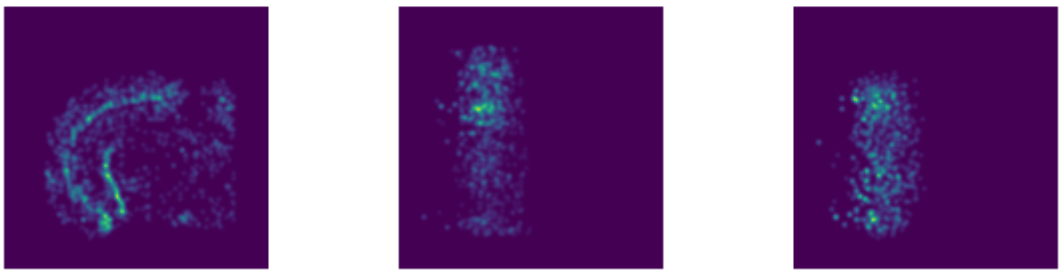


Real GEX Hs3st4

Real GEX (Tissue 1) - X-Y-Z-axis Hs3st4

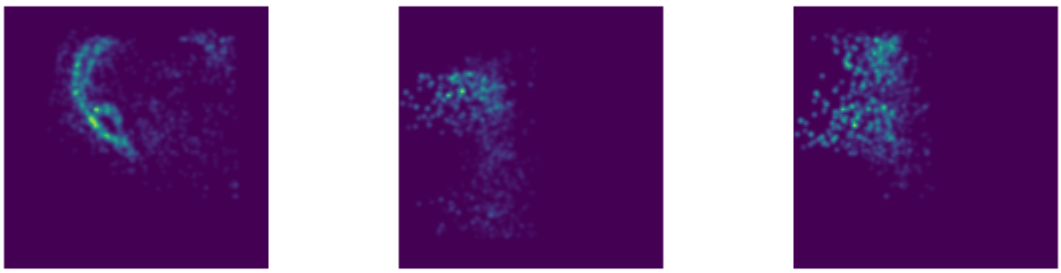


Real GEX (Tissue 2) - X-Y-Z-axis Hs3st4

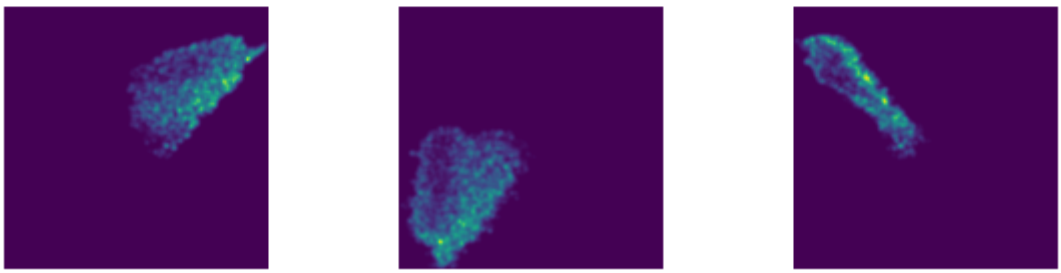


Real GEX (Tissue 3) - X-Y-Z-axis Hs3st4

Real GEX (Tissue 4) - X-Y-Z-axis Hs3st4

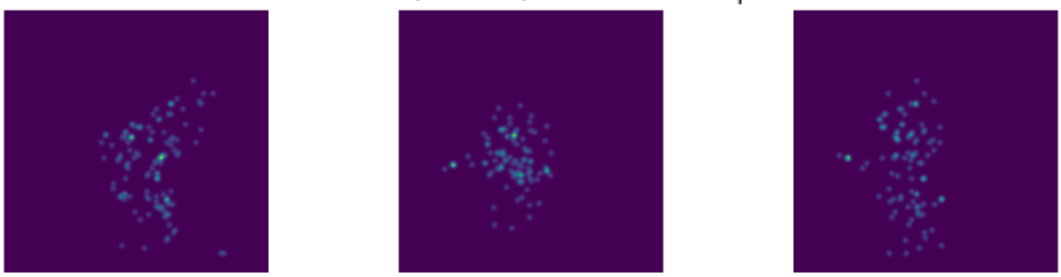


In-Silico GEX - X-Y-Z-axis Hs3st4

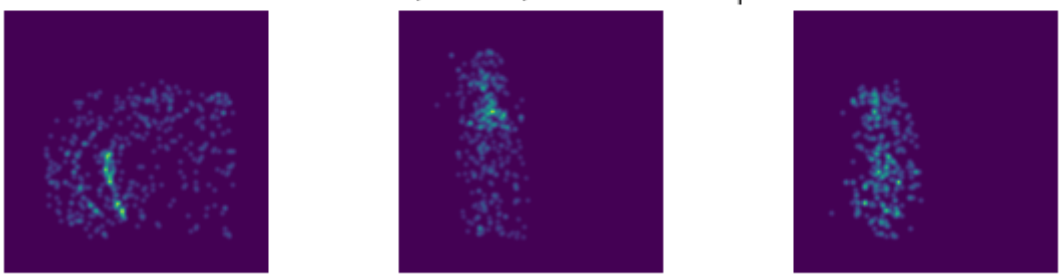


Real GEX Nrp1

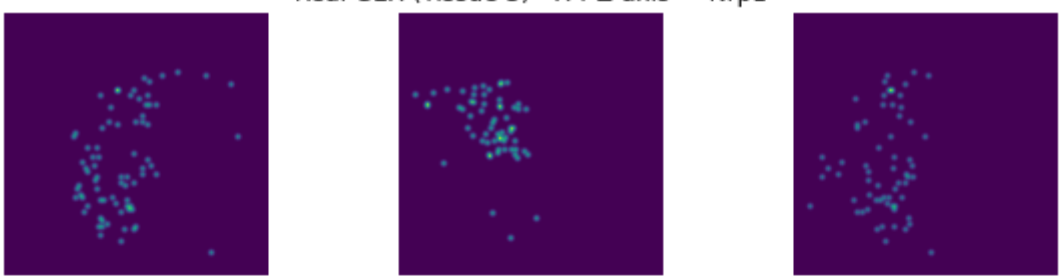
Real GEX (Tissue 1) - X-Y-Z-axis Nrp1



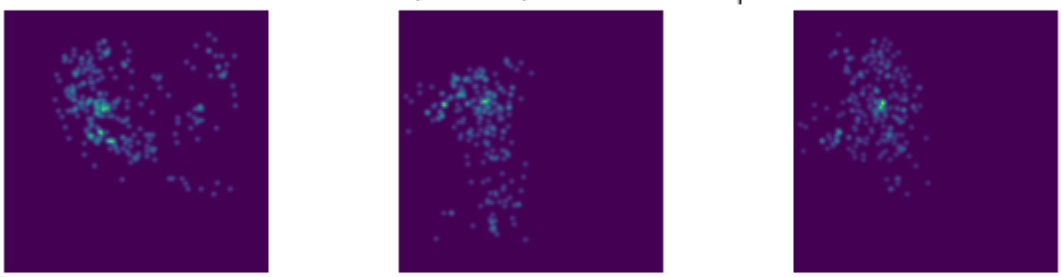
Real GEX (Tissue 2) - X-Y-Z-axis Nrp1



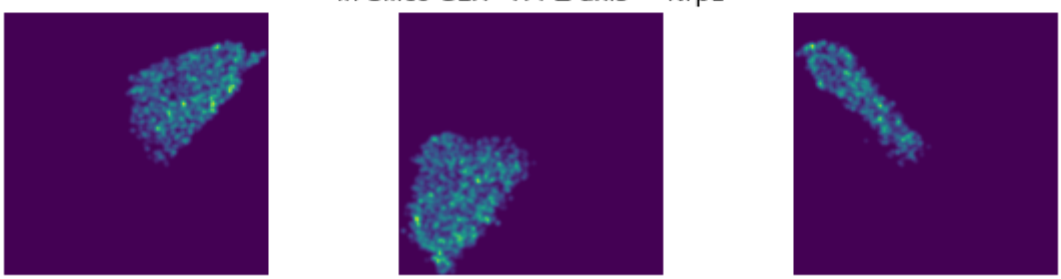
Real GEX (Tissue 3) - X-Y-Z-axis Nrp1



Real GEX (Tissue 4) - X-Y-Z-axis Nrp1

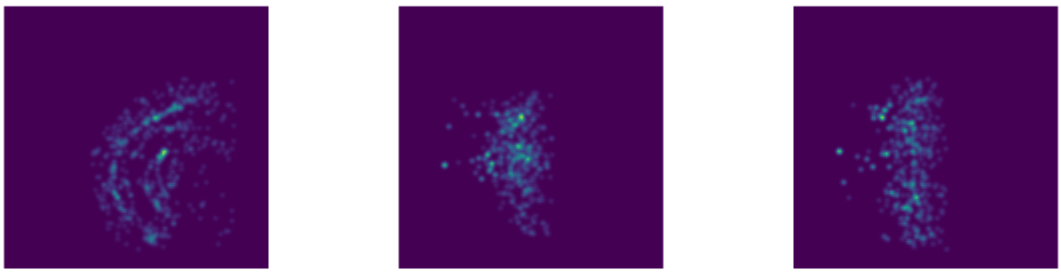


In-Silico GEX - X-Y-Z-axis Nrp1

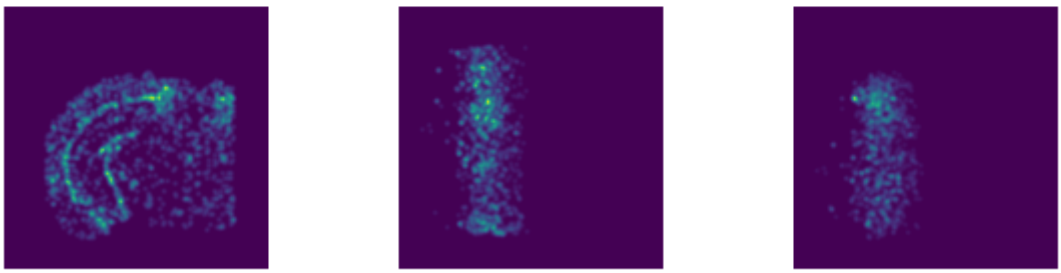


Real GEX Tle4

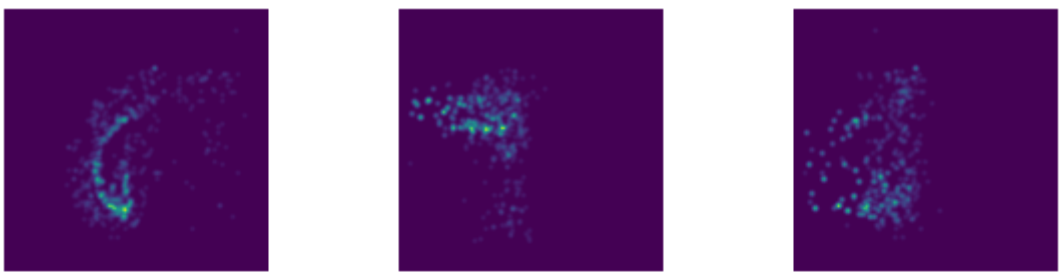
Real GEX (Tissue 1) - X-Y-Z-axis Tle4



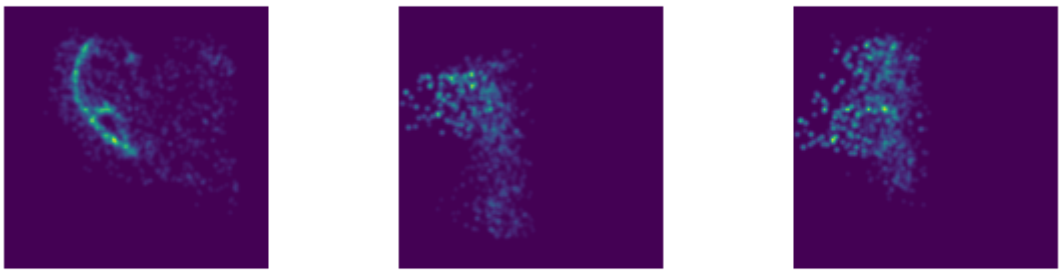
Real GEX (Tissue 2) - X-Y-Z-axis Tle4



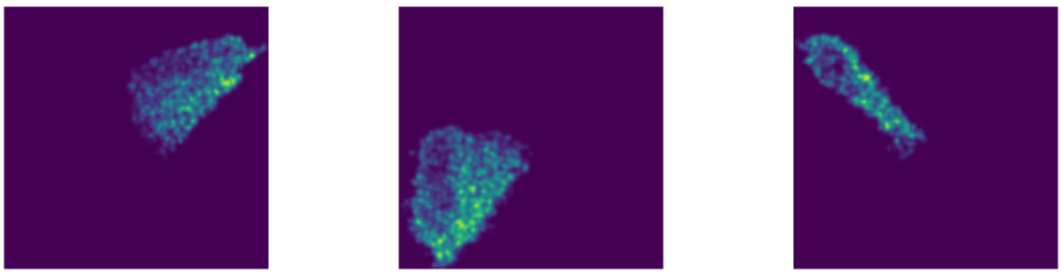
Real GEX (Tissue 3) - X-Y-Z-axis Tle4



Real GEX (Tissue 4) - X-Y-Z-axis Tle4

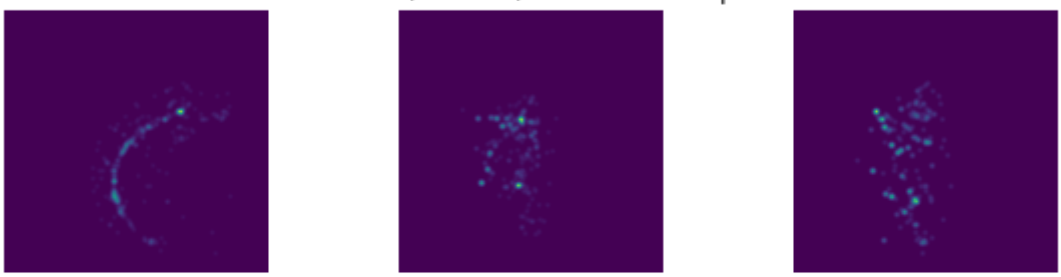


In-Silico GEX - X-Y-Z-axis Tle4

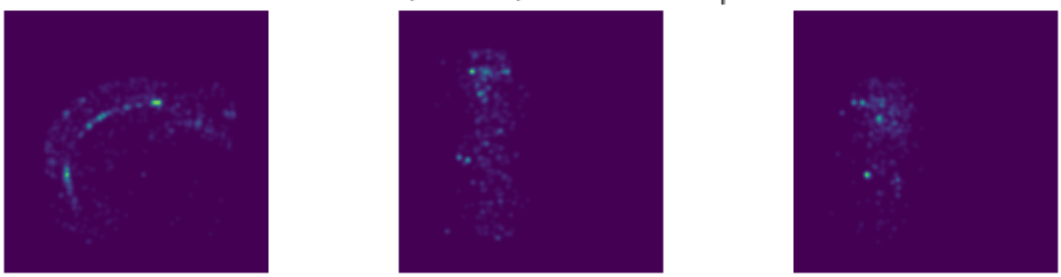


Real GEX Cplx3

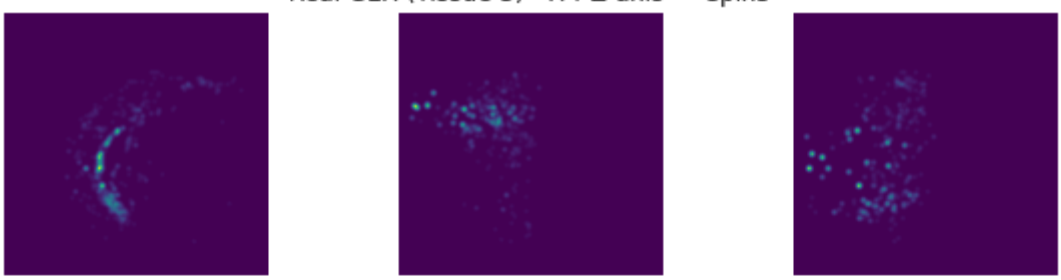
Real GEX (Tissue 1) - X-Y-Z-axis Cplx3



Real GEX (Tissue 2) - X-Y-Z-axis Cplx3

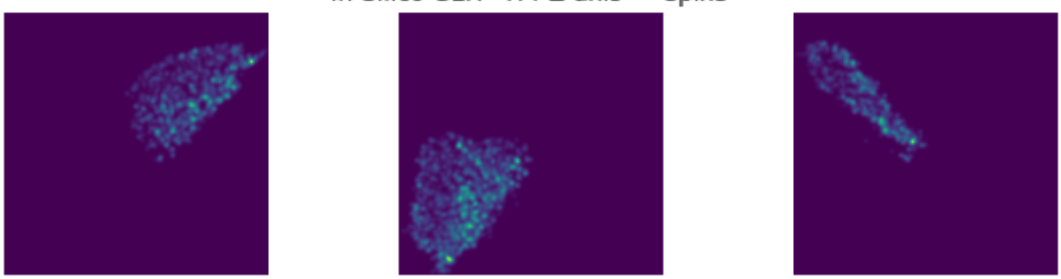


Real GEX (Tissue 3) - X-Y-Z-axis Cplx3



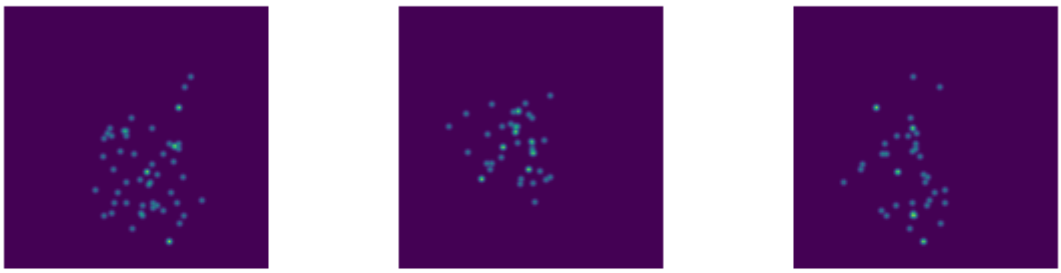
Real GEX (Tissue 4) - X-Y-Z-axis Cplx3

In-Silico GEX - X-Y-Z-axis Cplx3

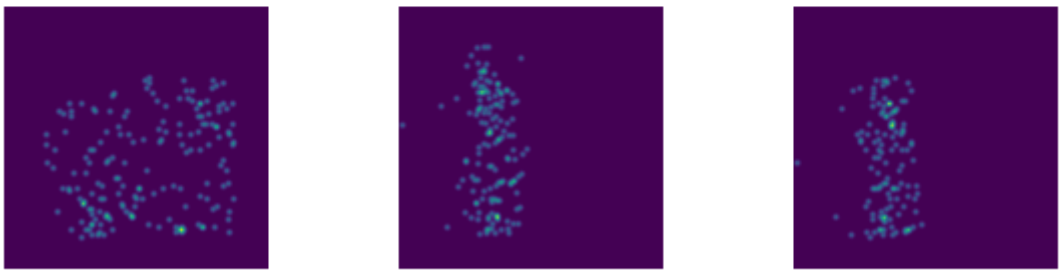


Real GEX Svil

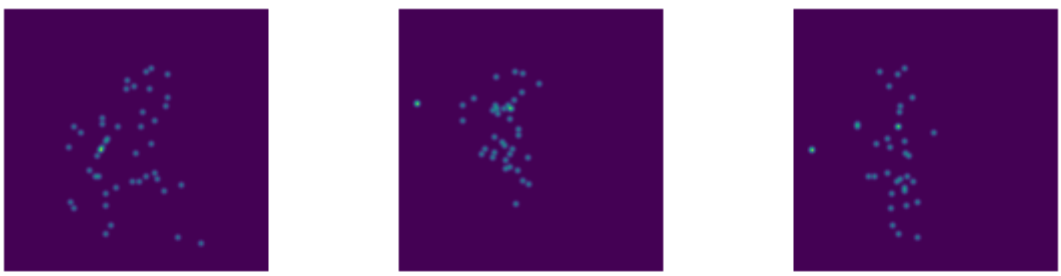
Real GEX (Tissue 1) - X-Y-Z-axis Svil



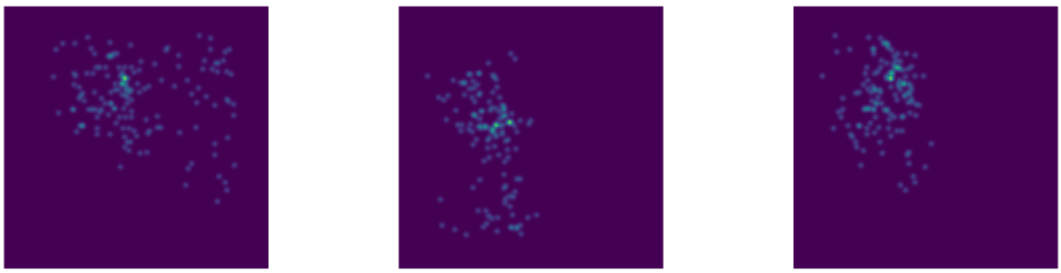
Real GEX (Tissue 2) - X-Y-Z-axis Svil



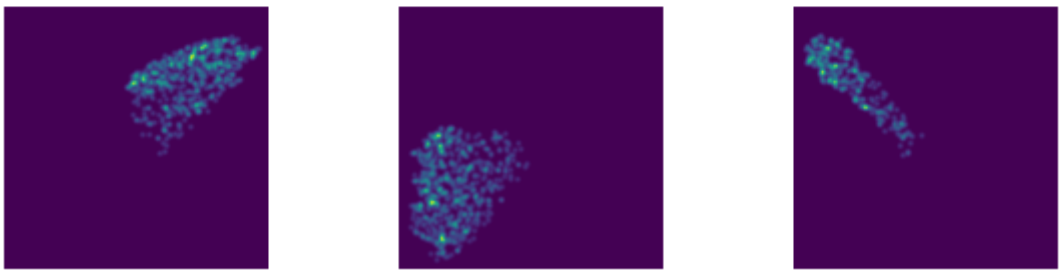
Real GEX (Tissue 3) - X-Y-Z-axis Svil



Real GEX (Tissue 4) - X-Y-Z-axis Svil



In-Silico GEX - X-Y-Z-axis Svil



Real GEX Fbxl7

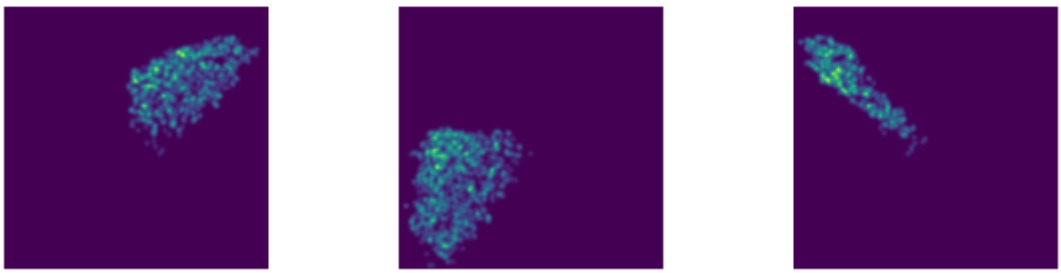
Real GEX (Tissue 1) - X-Y-Z-axis Fbxl7

Real GEX (Tissue 2) - X-Y-Z-axis Fbxl7

Real GEX (Tissue 3) - X-Y-Z-axis Fbxl7

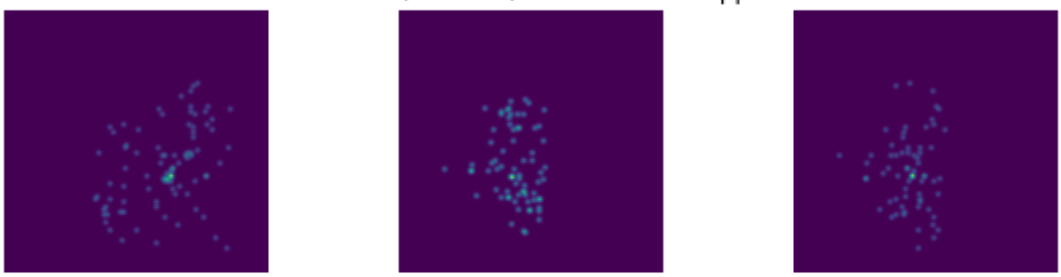
Real GEX (Tissue 4) - X-Y-Z-axis Fbxl7

In-Silico GEX - X-Y-Z-axis Fbxl7

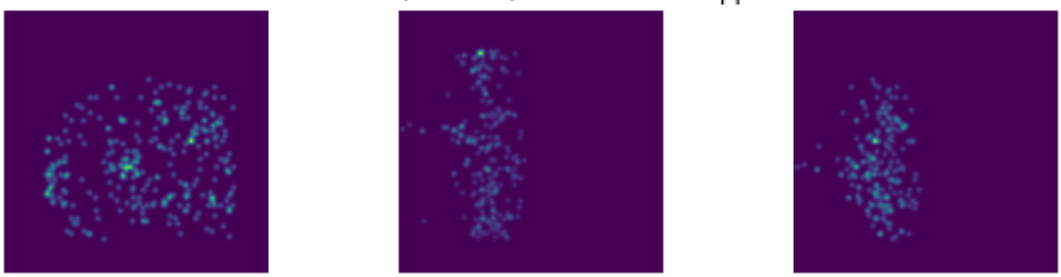


Real GEX Inpp4b

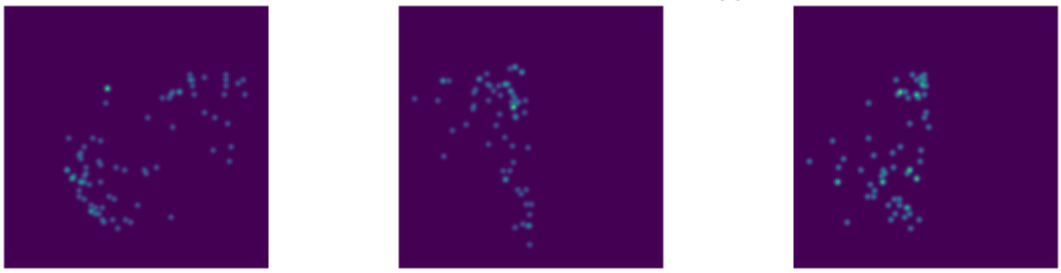
Real GEX (Tissue 1) - X-Y-Z-axis Inpp4b



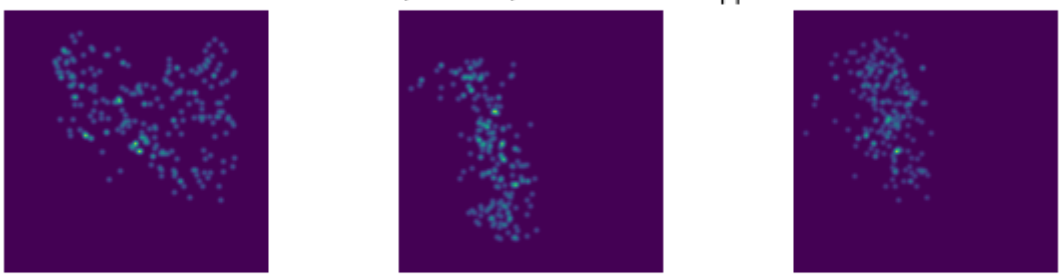
Real GEX (Tissue 2) - X-Y-Z-axis Inpp4b



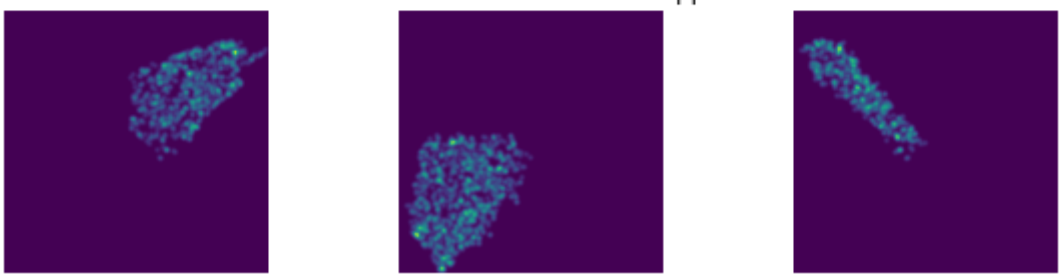
Real GEX (Tissue 3) - X-Y-Z-axis Inpp4b



Real GEX (Tissue 4) - X-Y-Z-axis Inpp4b

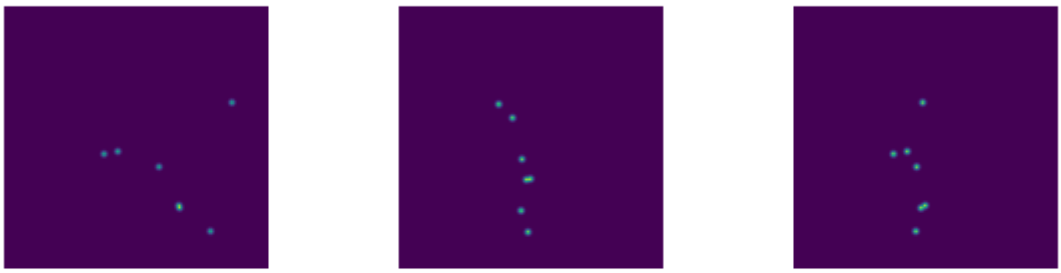


In-Silico GEX - X-Y-Z-axis Inpp4b

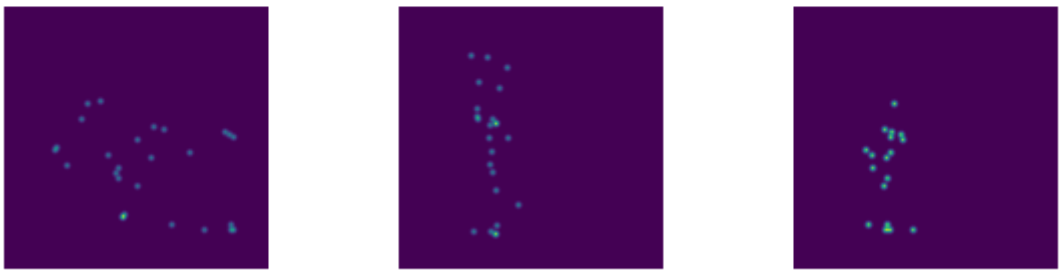


Real GEX Car3

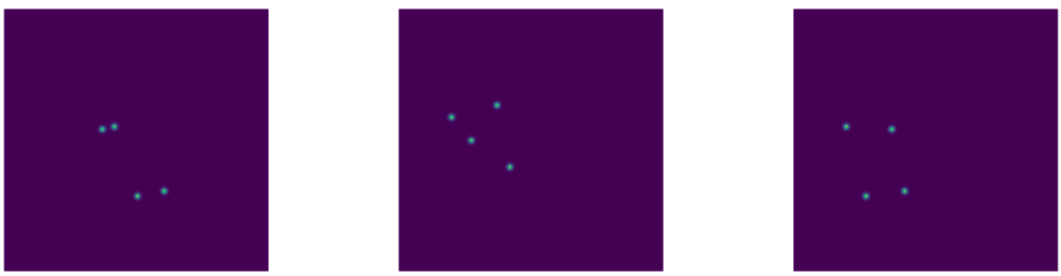
Real GEX (Tissue 1) - X-Y-Z-axis Car3



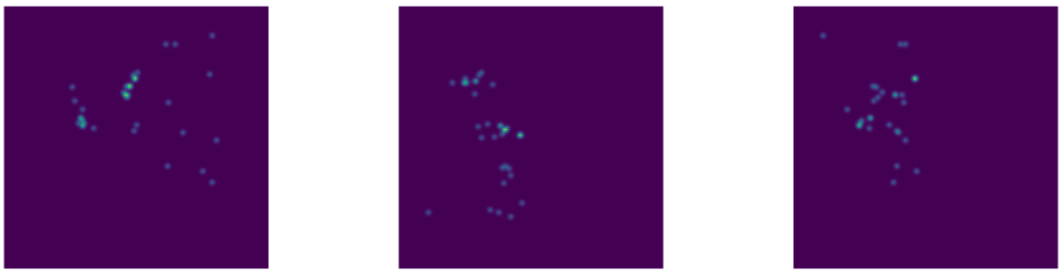
Real GEX (Tissue 2) - X-Y-Z-axis Car3



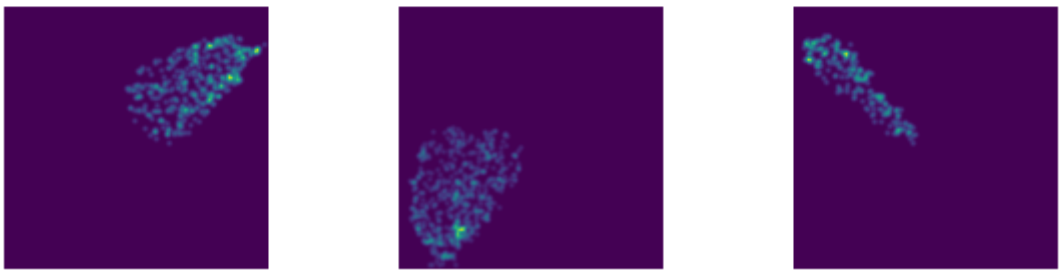
Real GEX (Tissue 3) - X-Y-Z-axis Car3



Real GEX (Tissue 4) - X-Y-Z-axis Car3



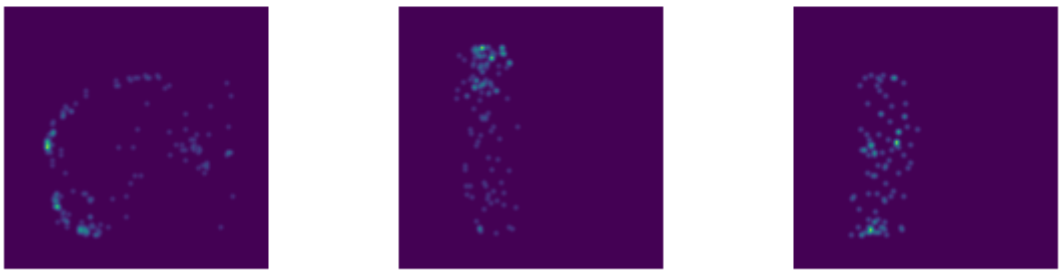
In-Silico GEX - X-Y-Z-axis Car3



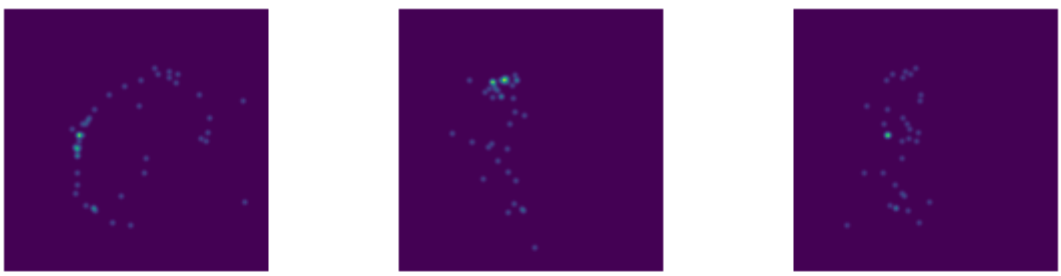
Real GEX Otof

Real GEX (Tissue 1) - X-Y-Z-axis Otof

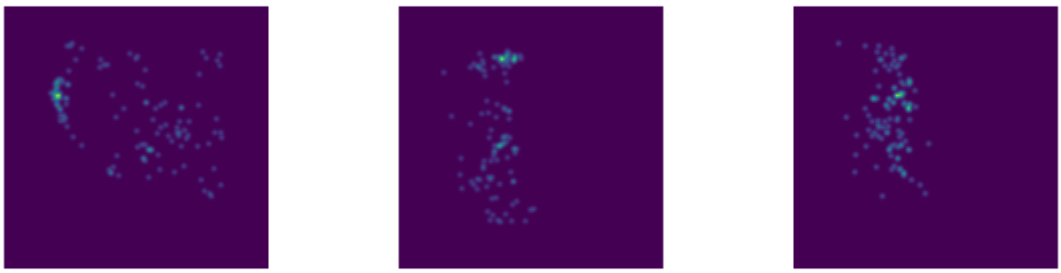
Real GEX (Tissue 2) - X-Y-Z-axis Otof



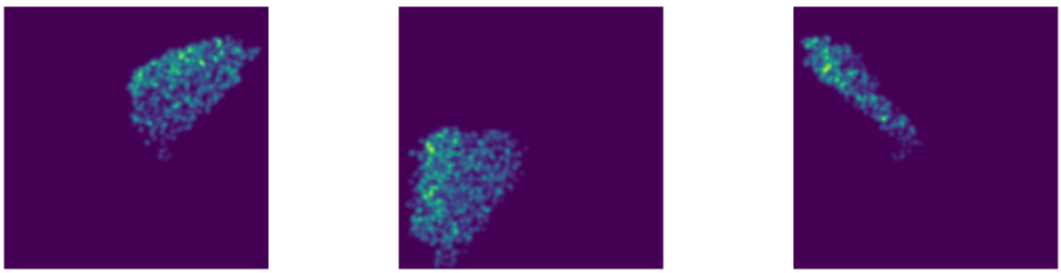
Real GEX (Tissue 3) - X-Y-Z-axis Otof



Real GEX (Tissue 4) - X-Y-Z-axis Otof

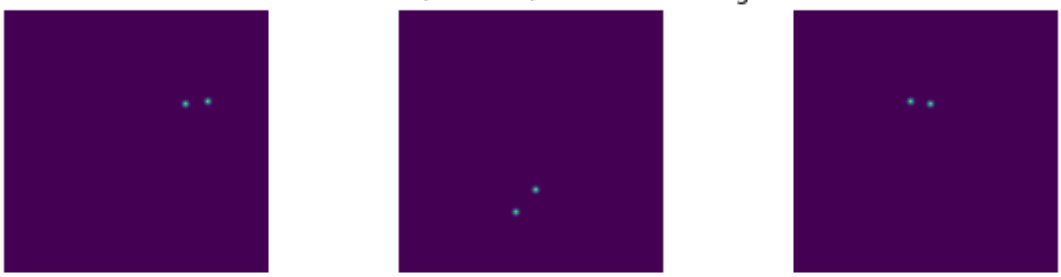


In-Silico GEX - X-Y-Z-axis Otof

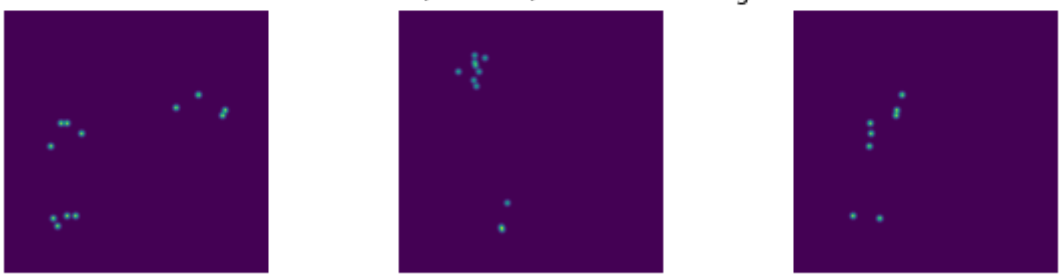


Real GEX Hgf

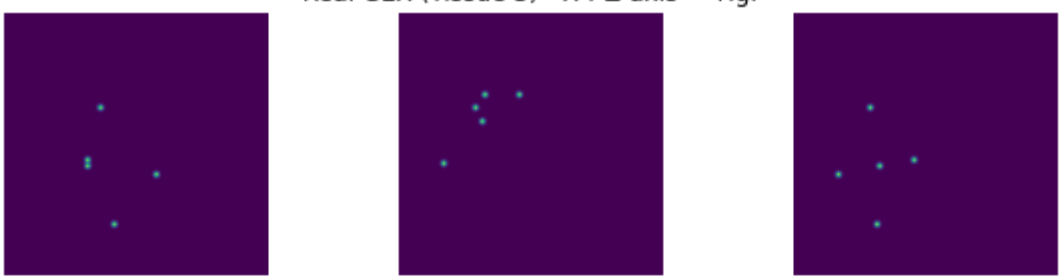
Real GEX (Tissue 1) - X-Y-Z-axis Hgf



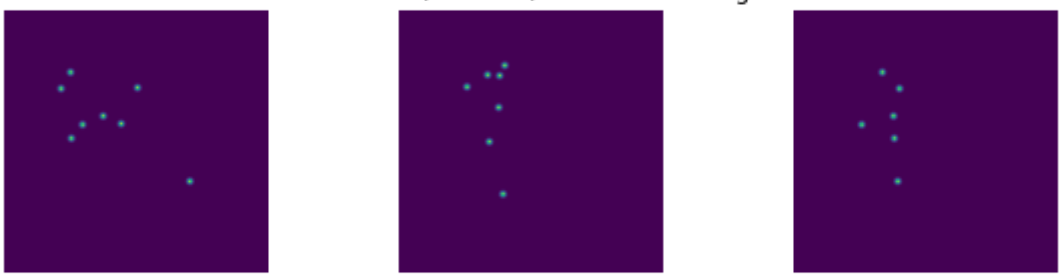
Real GEX (Tissue 2) - X-Y-Z-axis Hgf



Real GEX (Tissue 3) - X-Y-Z-axis Hgf



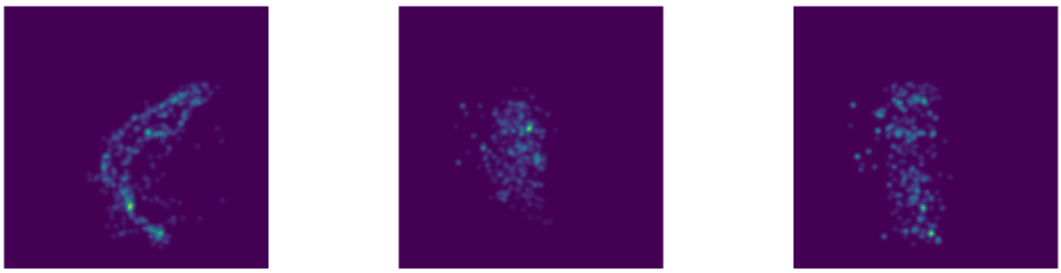
Real GEX (Tissue 4) - X-Y-Z-axis Hgf



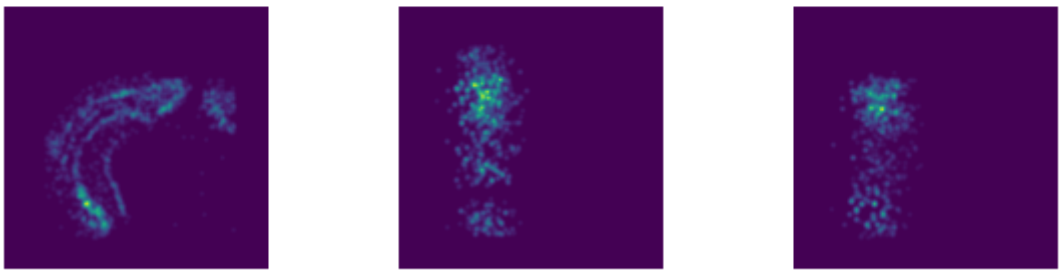
In-Silico GEX - X-Y-Z-axis Hgf

Real GEX Fezf2

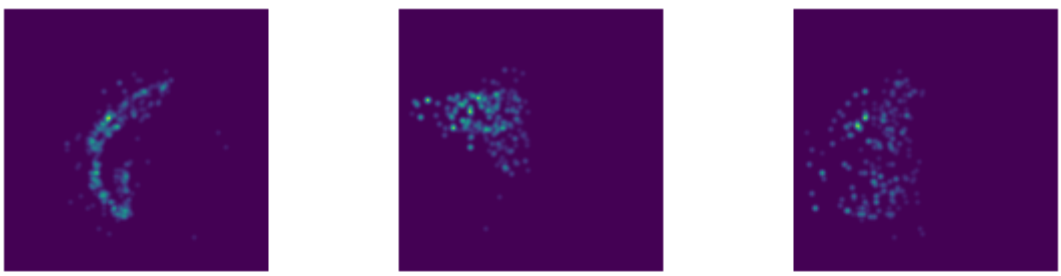
Real GEX (Tissue 1) - X-Y-Z-axis Fezf2



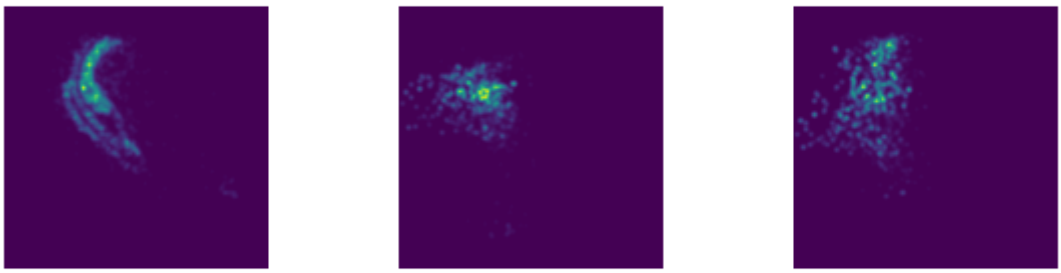
Real GEX (Tissue 2) - X-Y-Z-axis Fezf2



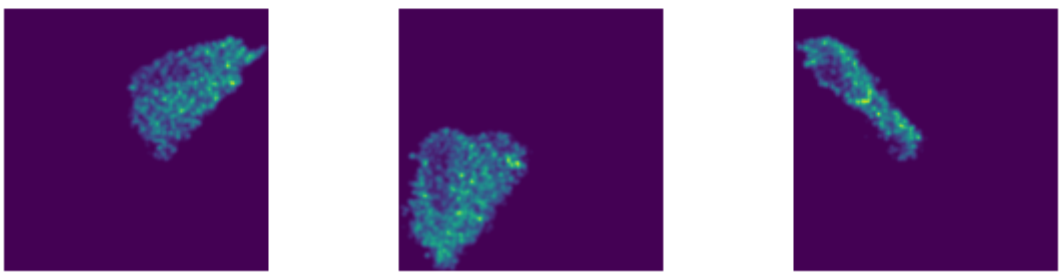
Real GEX (Tissue 3) - X-Y-Z-axis Fezf2



Real GEX (Tissue 4) - X-Y-Z-axis Fezf2

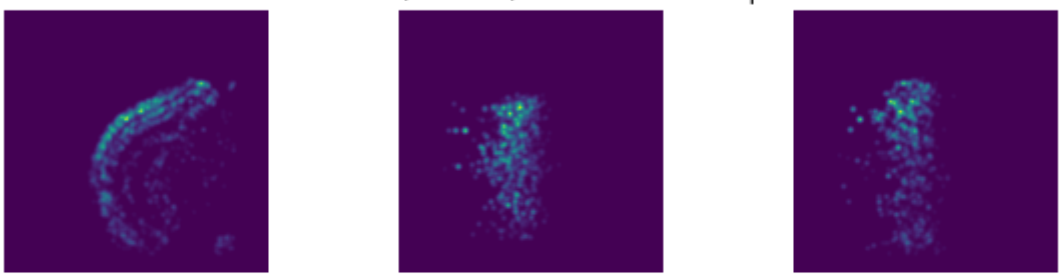


In-Silico GEX - X-Y-Z-axis Fezf2

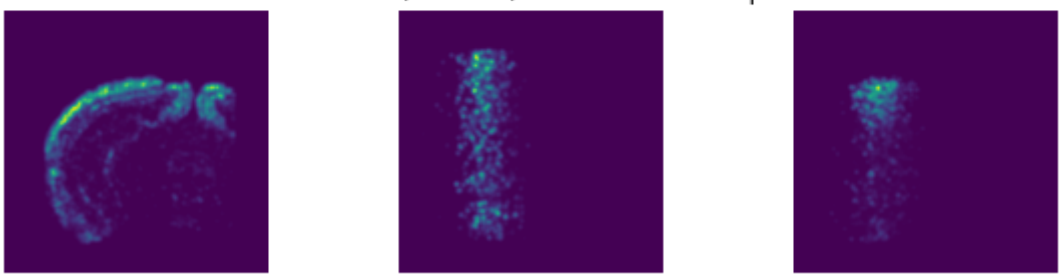


Real GEX Lamp5

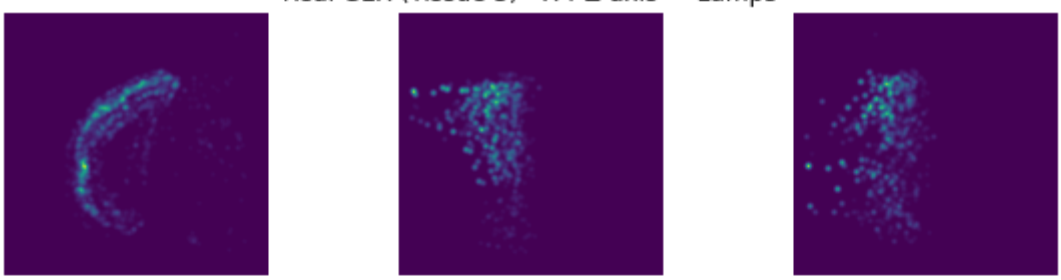
Real GEX (Tissue 1) - X-Y-Z-axis Lamp5



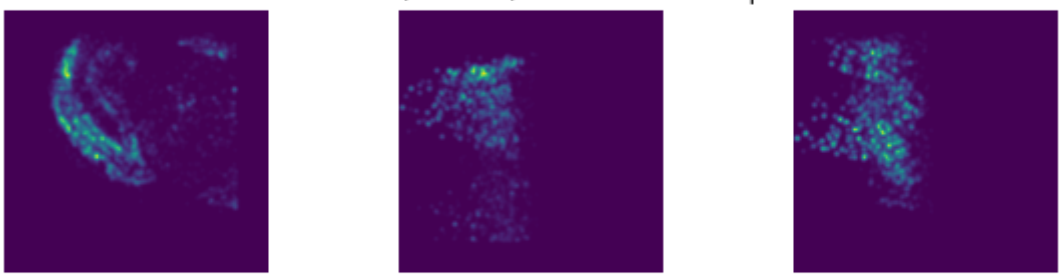
Real GEX (Tissue 2) - X-Y-Z-axis Lamp5



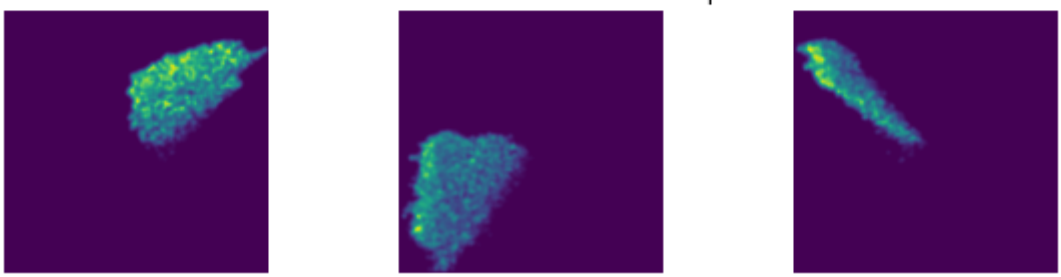
Real GEX (Tissue 3) - X-Y-Z-axis Lamp5



Real GEX (Tissue 4) - X-Y-Z-axis Lamp5

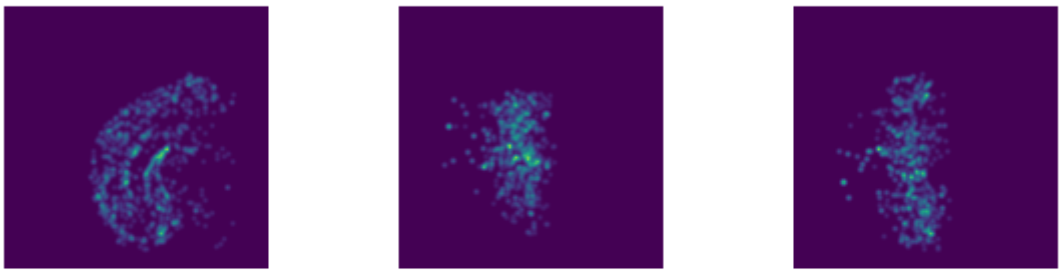


In-Silico GEX - X-Y-Z-axis Lamp5

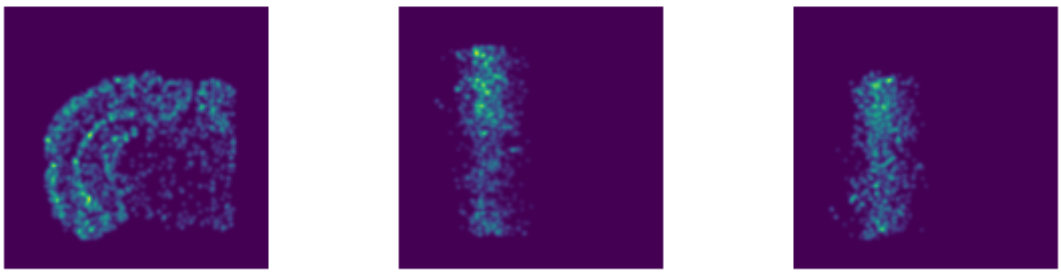


Real GEX Dab1

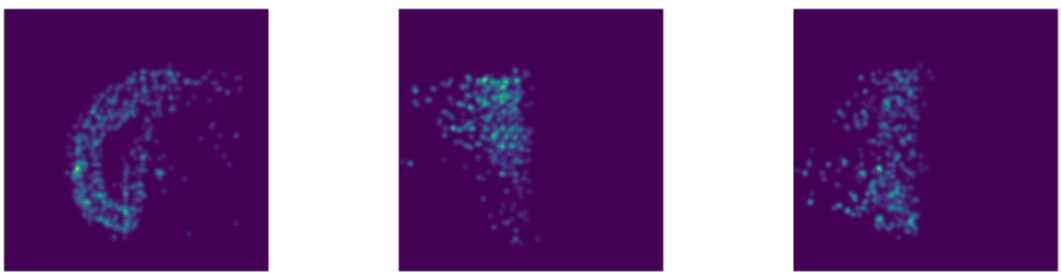
Real GEX (Tissue 1) - X-Y-Z-axis Dab1



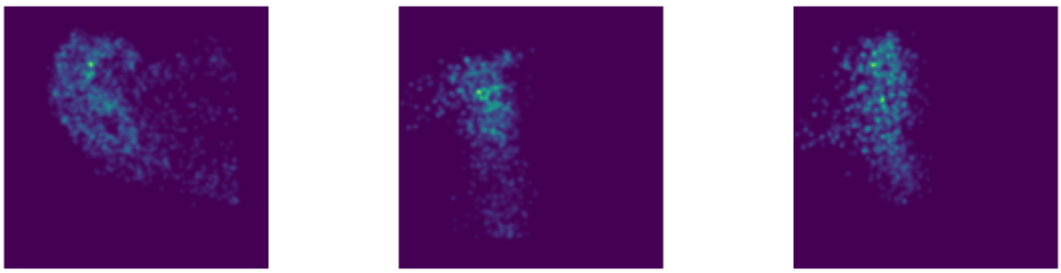
Real GEX (Tissue 2) - X-Y-Z-axis Dab1



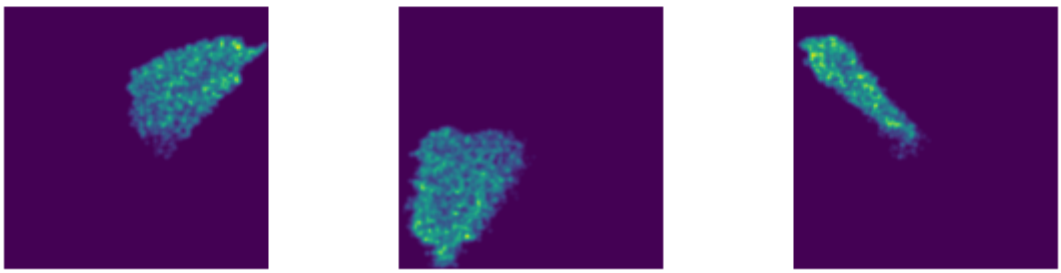
Real GEX (Tissue 3) - X-Y-Z-axis Dab1



Real GEX (Tissue 4) - X-Y-Z-axis Dab1

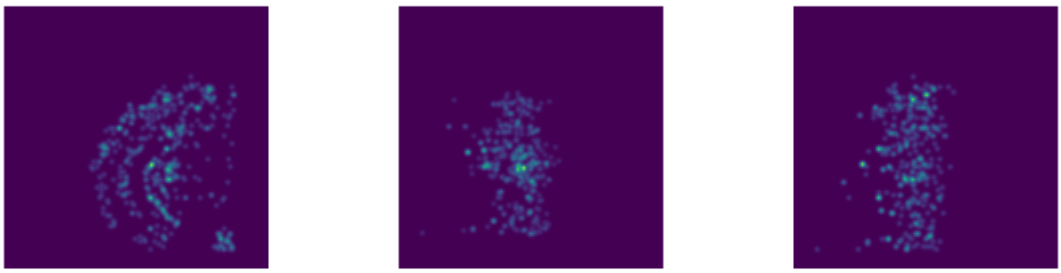


In-Silico GEX - X-Y-Z-axis Dab1

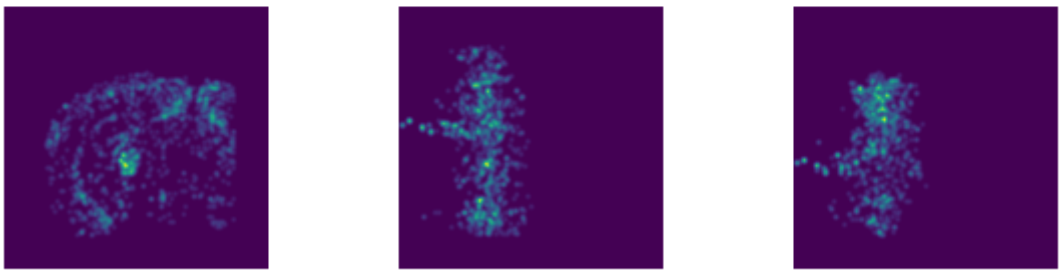


Real GEX Zmat4

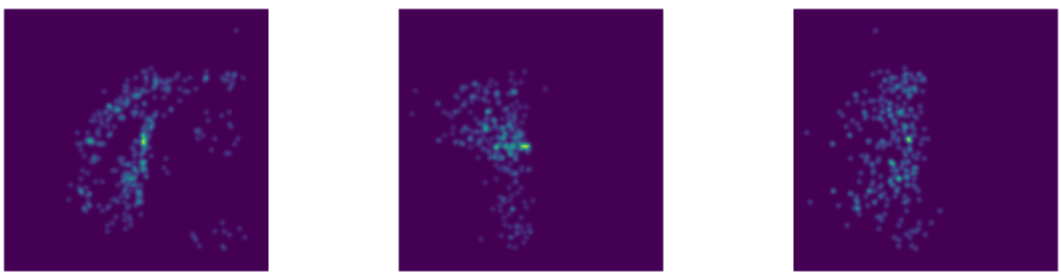
Real GEX (Tissue 1) - X-Y-Z-axis Zmat4



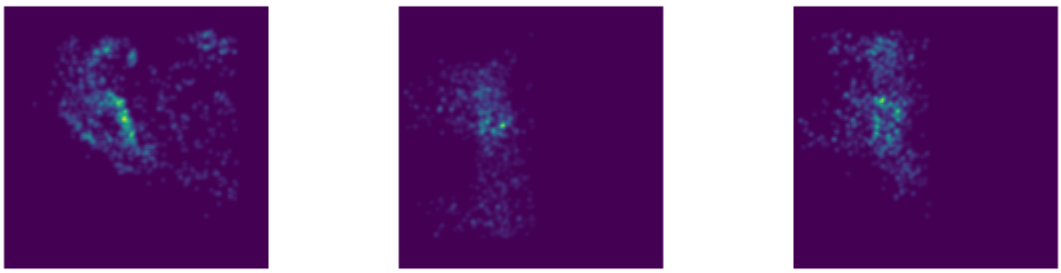
Real GEX (Tissue 2) - X-Y-Z-axis Zmat4



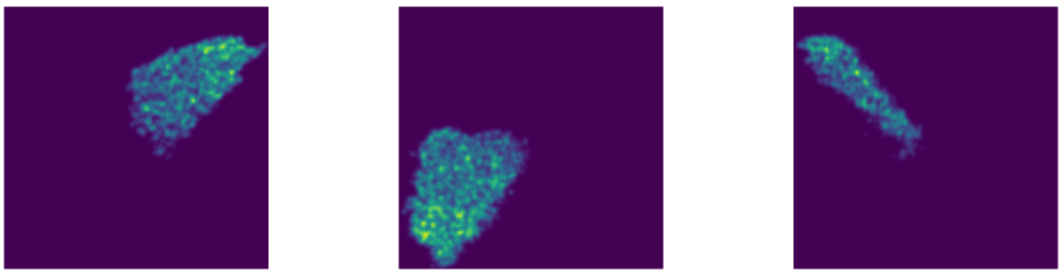
Real GEX (Tissue 3) - X-Y-Z-axis Zmat4



Real GEX (Tissue 4) - X-Y-Z-axis Zmat4

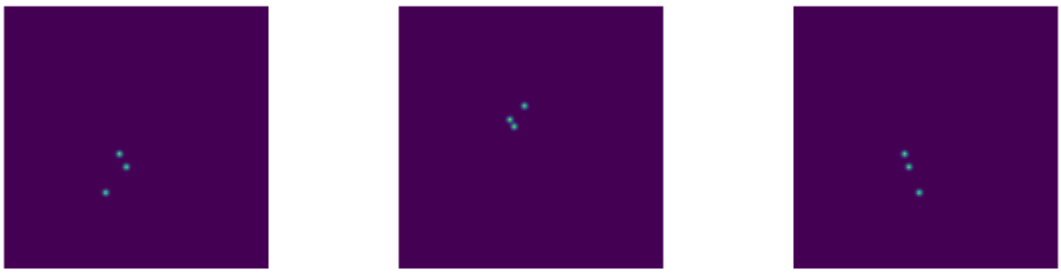


In-Silico GEX - X-Y-Z-axis Zmat4



Real GEX Hs6st3

Real GEX (Tissue 1) - X-Y-Z-axis Hs6st3

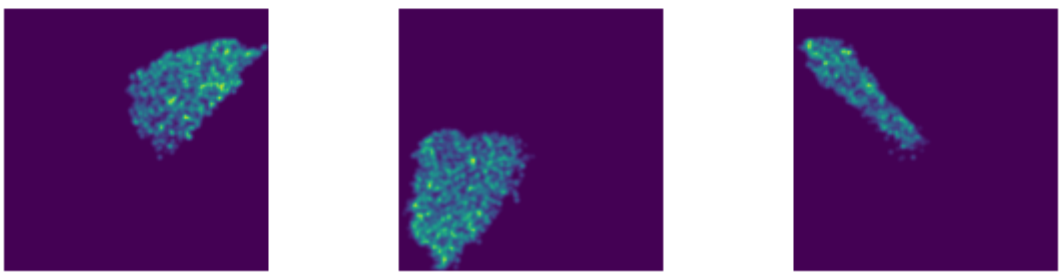


Real GEX (Tissue 2) - X-Y-Z-axis Hs6st3

Real GEX (Tissue 3) - X-Y-Z-axis Hs6st3

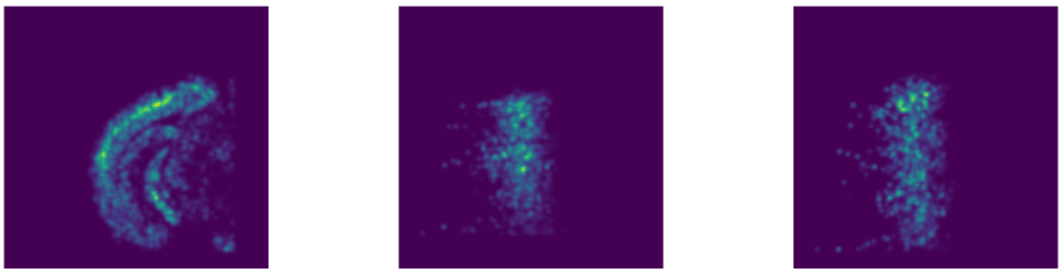
Real GEX (Tissue 4) - X-Y-Z-axis Hs6st3

In-Silico GEX - X-Y-Z-axis Hs6st3

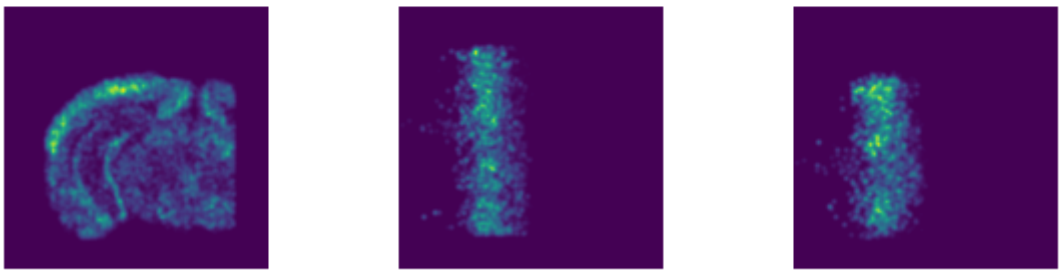


Real GEX Nrsn1

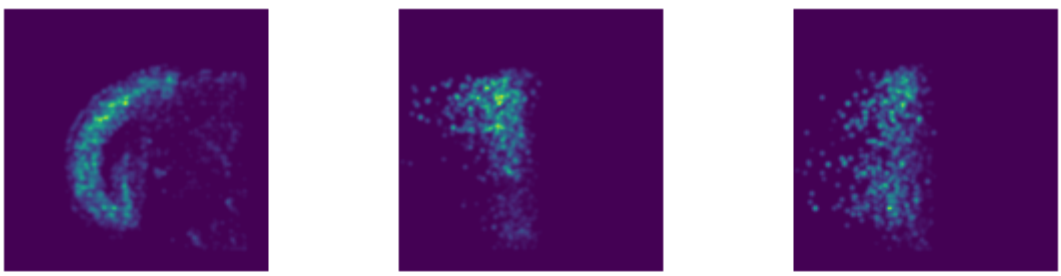
Real GEX (Tissue 1) - X-Y-Z-axis Nrsn1



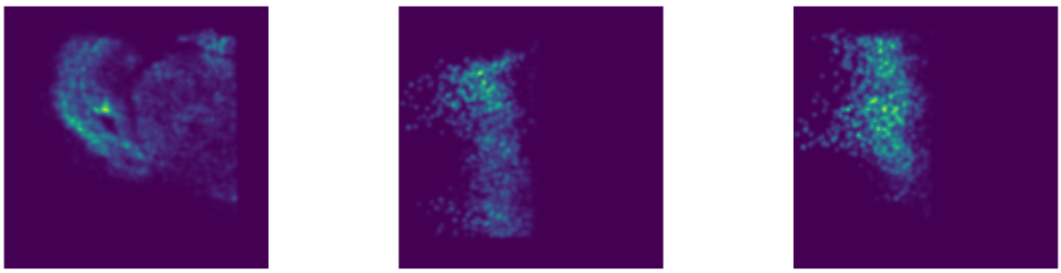
Real GEX (Tissue 2) - X-Y-Z-axis Nrsn1



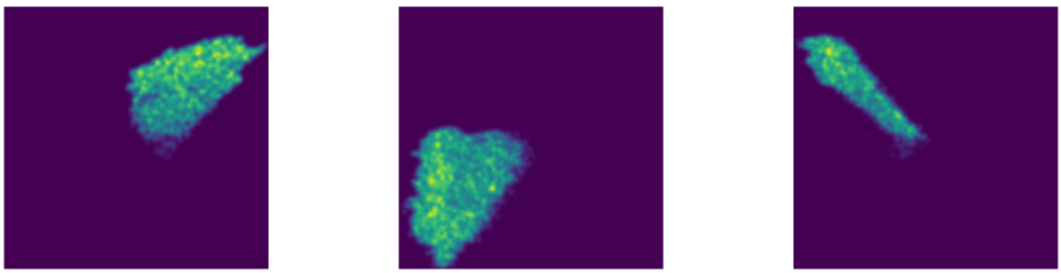
Real GEX (Tissue 3) - X-Y-Z-axis Nrsn1



Real GEX (Tissue 4) - X-Y-Z-axis Nrsn1

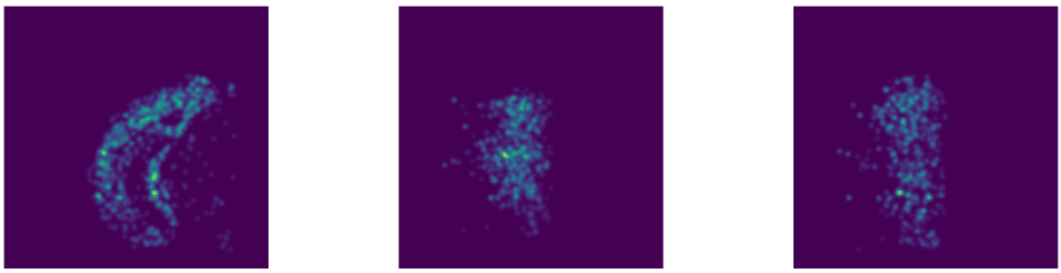


In-Silico GEX - X-Y-Z-axis Nrsn1

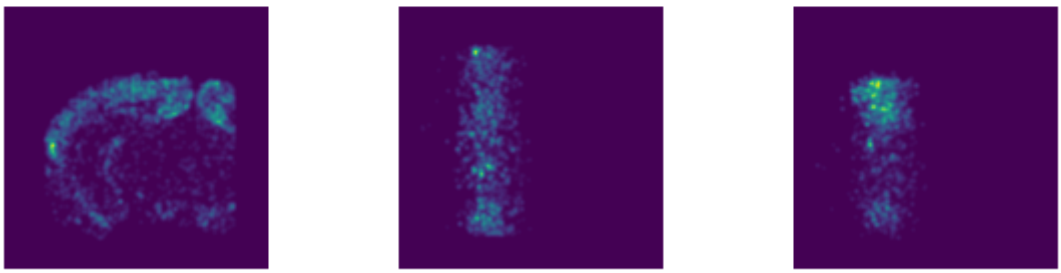


Real GEX Satb1

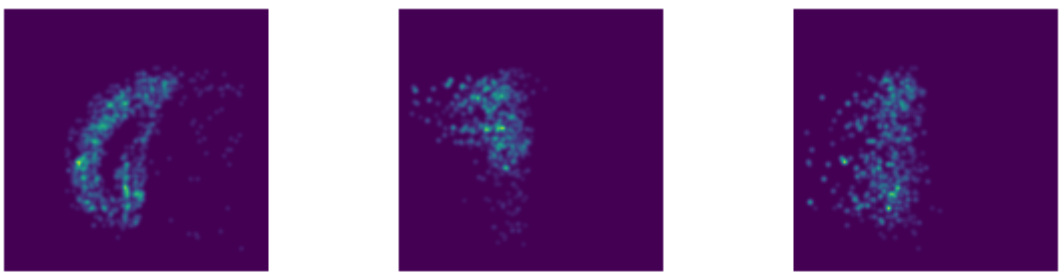
Real GEX (Tissue 1) - X-Y-Z-axis Satb1



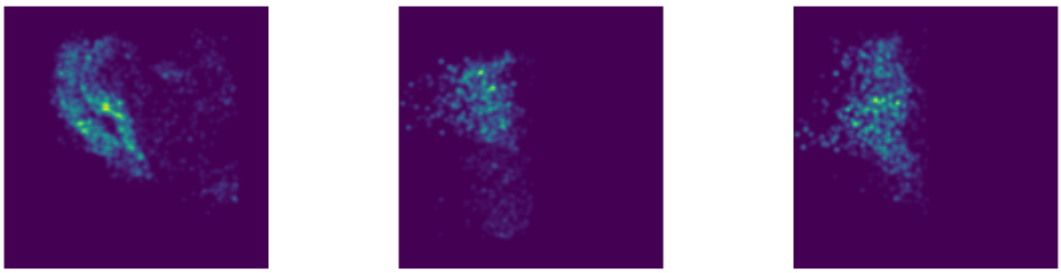
Real GEX (Tissue 2) - X-Y-Z-axis Satb1



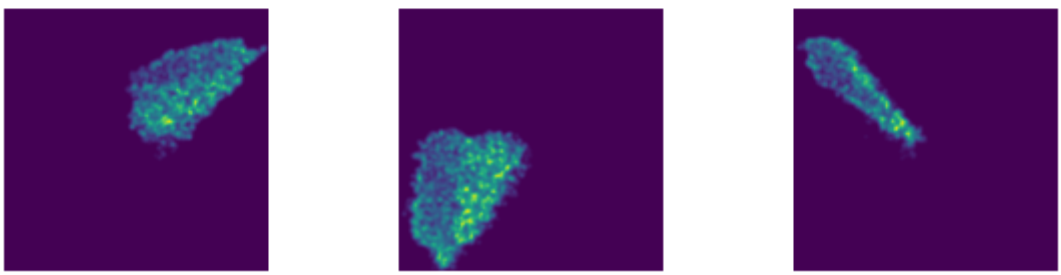
Real GEX (Tissue 3) - X-Y-Z-axis Satb1



Real GEX (Tissue 4) - X-Y-Z-axis Satb1

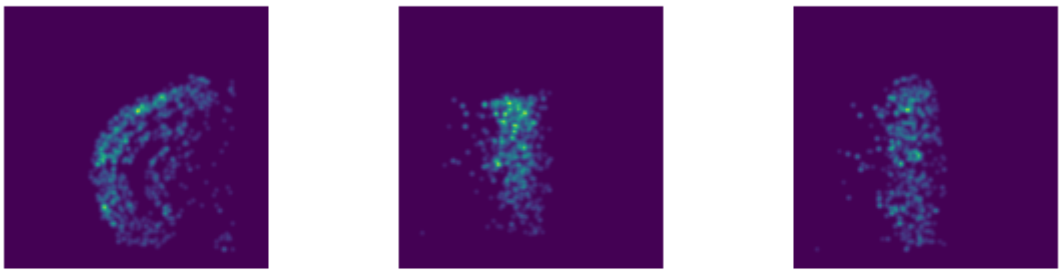


In-Silico GEX - X-Y-Z-axis Satb1

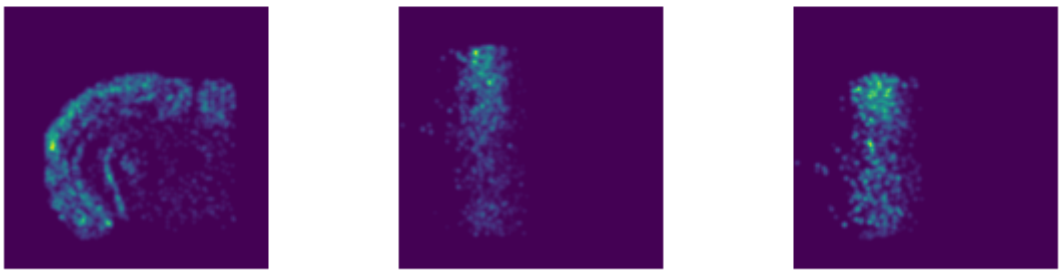


Real GEX Camk4

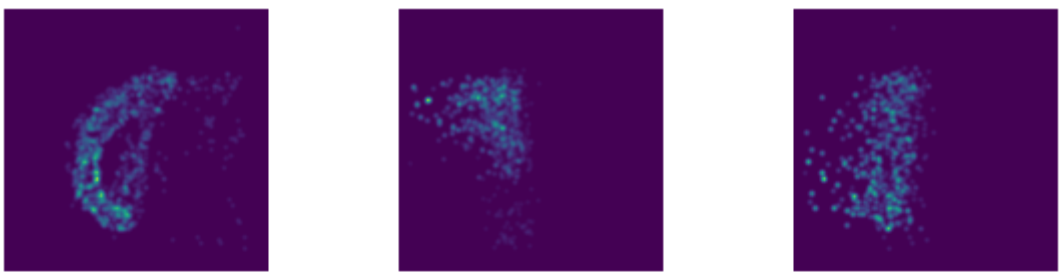
Real GEX (Tissue 1) - X-Y-Z-axis Camk4



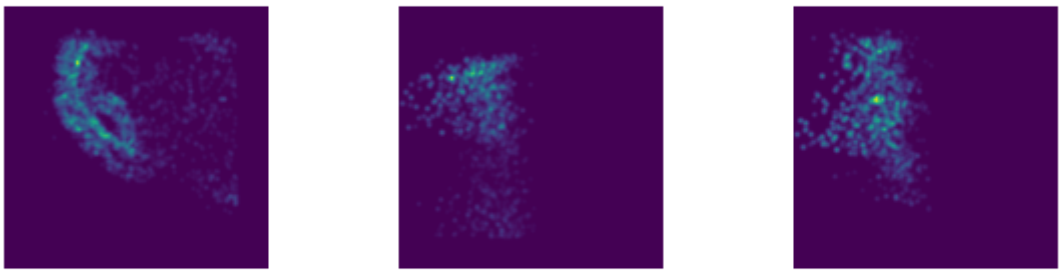
Real GEX (Tissue 2) - X-Y-Z-axis Camk4



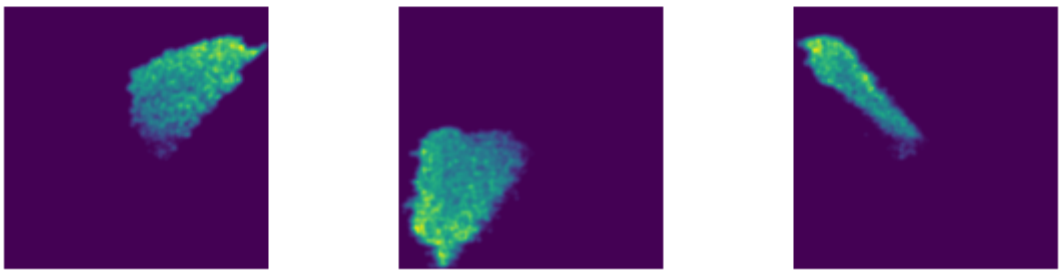
Real GEX (Tissue 3) - X-Y-Z-axis Camk4



Real GEX (Tissue 4) - X-Y-Z-axis Camk4

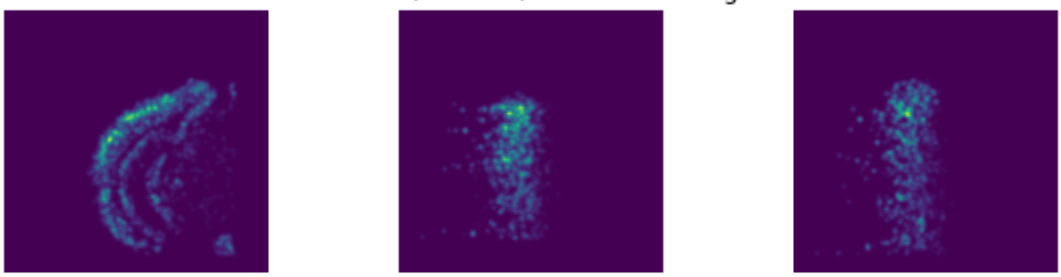


In-Silico GEX - X-Y-Z-axis Camk4

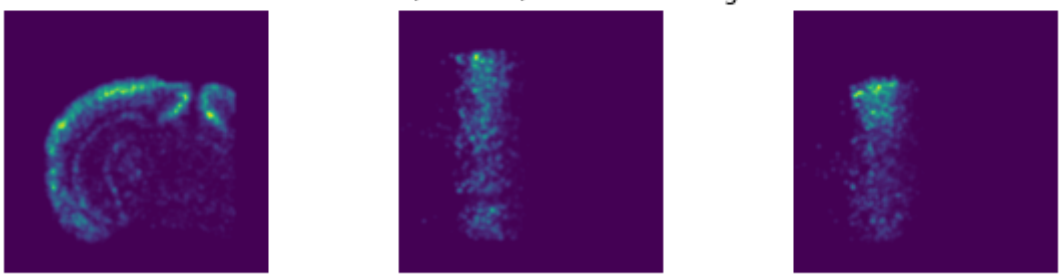


Real GEX Rgs4

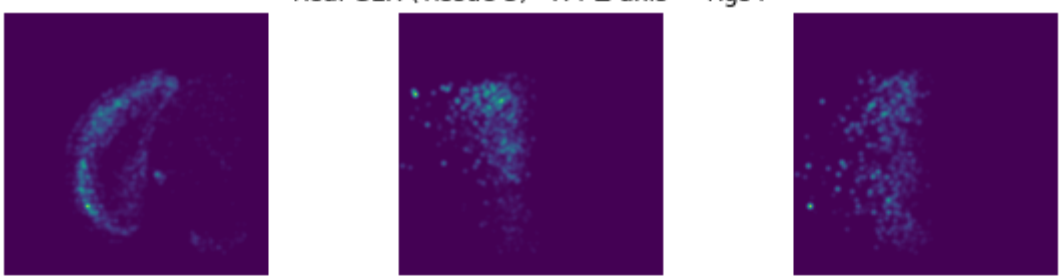
Real GEX (Tissue 1) - X-Y-Z-axis Rgs4



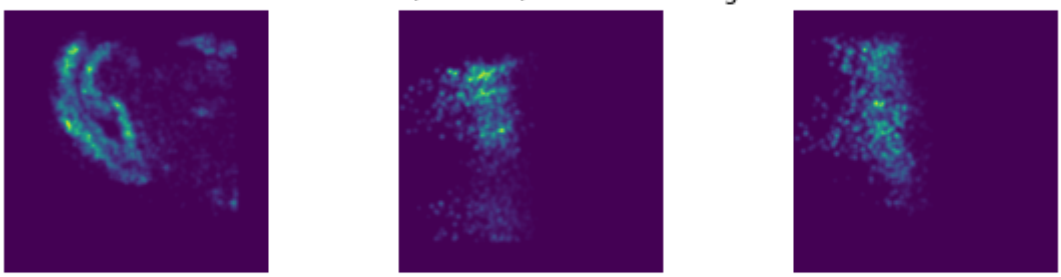
Real GEX (Tissue 2) - X-Y-Z-axis Rgs4



Real GEX (Tissue 3) - X-Y-Z-axis Rgs4



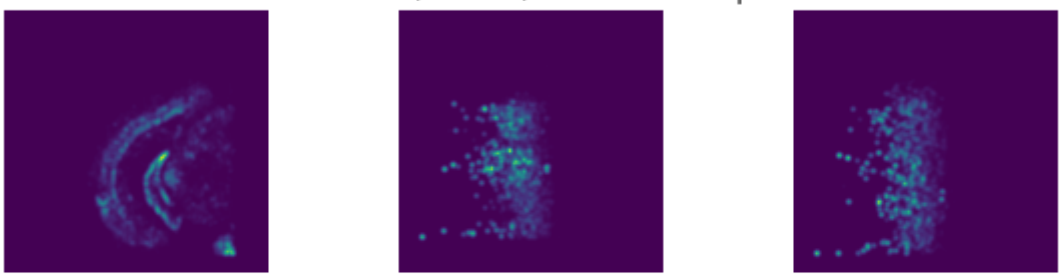
Real GEX (Tissue 4) - X-Y-Z-axis Rgs4



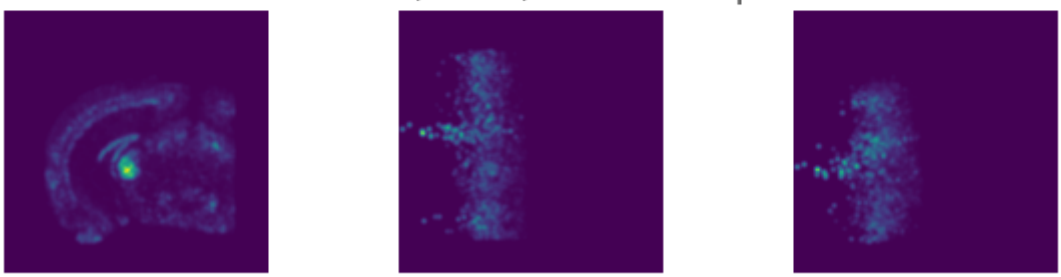
In-Silico GEX - X-Y-Z-axis Rgs4

Real GEX Pcp4

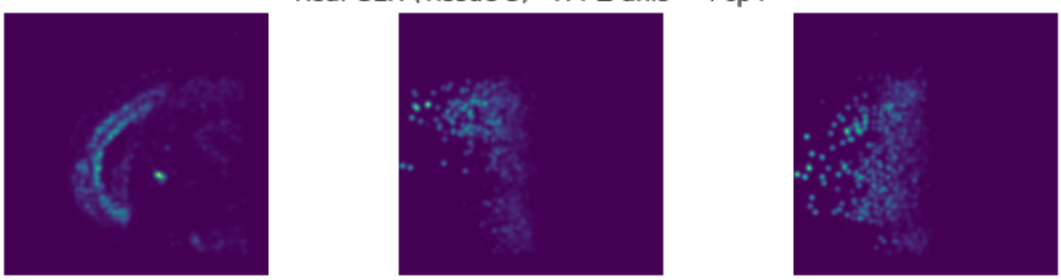
Real GEX (Tissue 1) - X-Y-Z-axis Pcp4



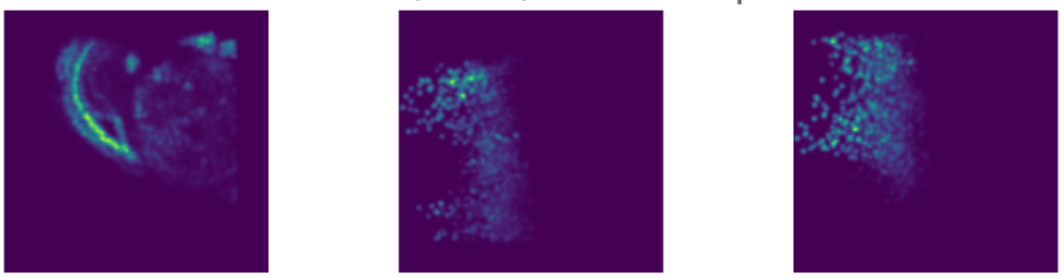
Real GEX (Tissue 2) - X-Y-Z-axis Pcp4



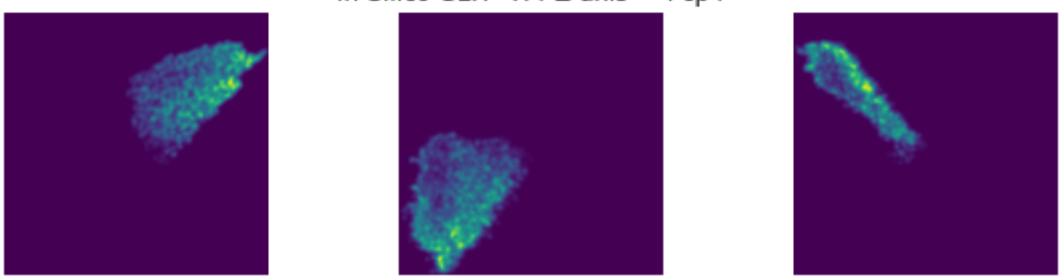
Real GEX (Tissue 3) - X-Y-Z-axis Pcp4



Real GEX (Tissue 4) - X-Y-Z-axis Pcp4

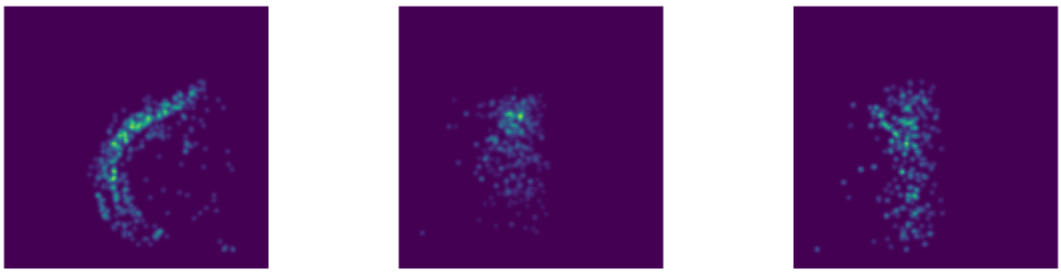


In-Silico GEX - X-Y-Z-axis Pcp4

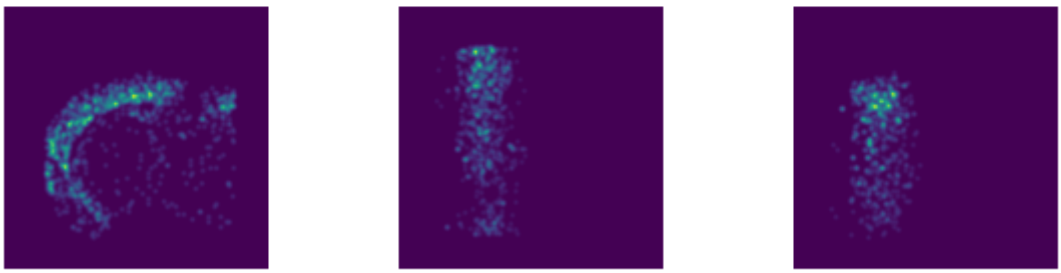


Real GEX Hs3st2

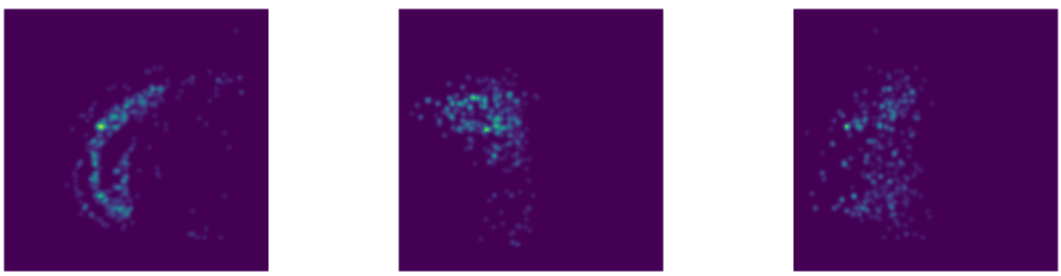
Real GEX (Tissue 1) - X-Y-Z-axis Hs3st2



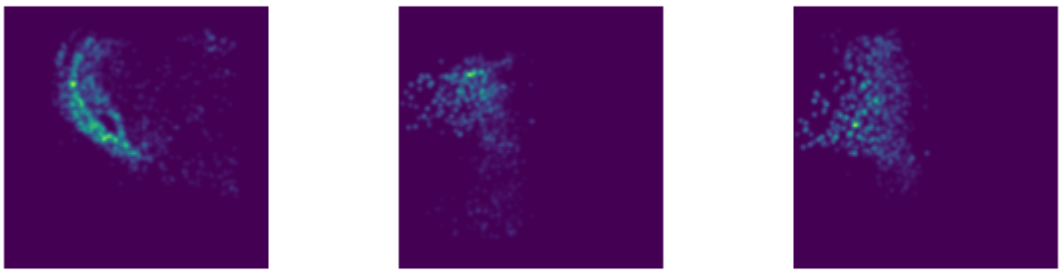
Real GEX (Tissue 2) - X-Y-Z-axis Hs3st2



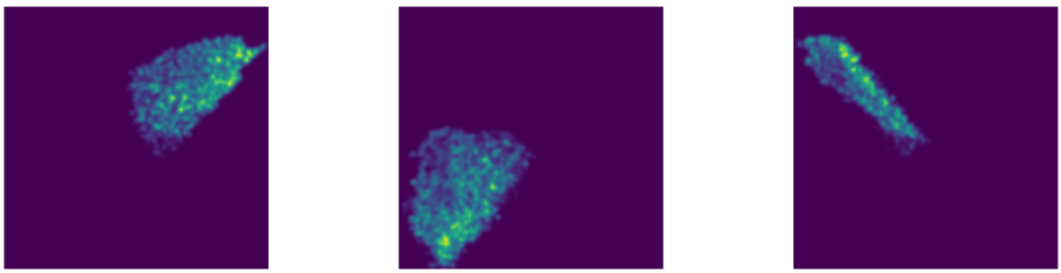
Real GEX (Tissue 3) - X-Y-Z-axis Hs3st2



Real GEX (Tissue 4) - X-Y-Z-axis Hs3st2

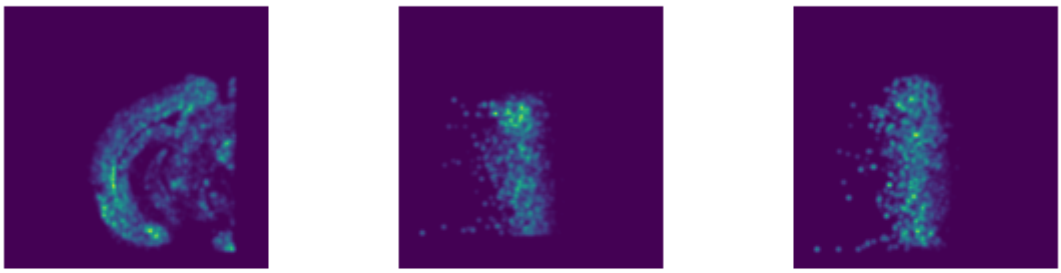


In-Silico GEX - X-Y-Z-axis Hs3st2

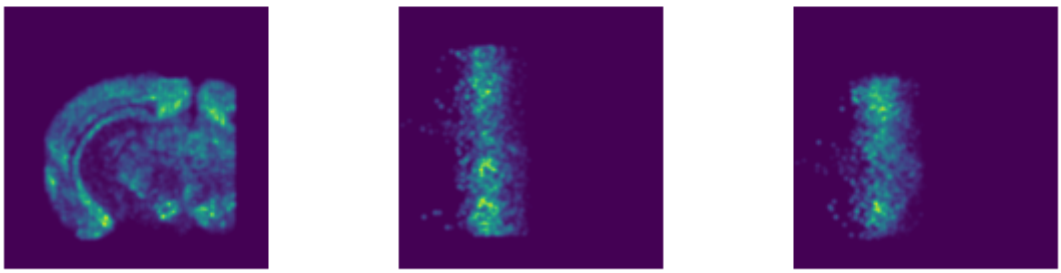


Real GEX Tmsb10

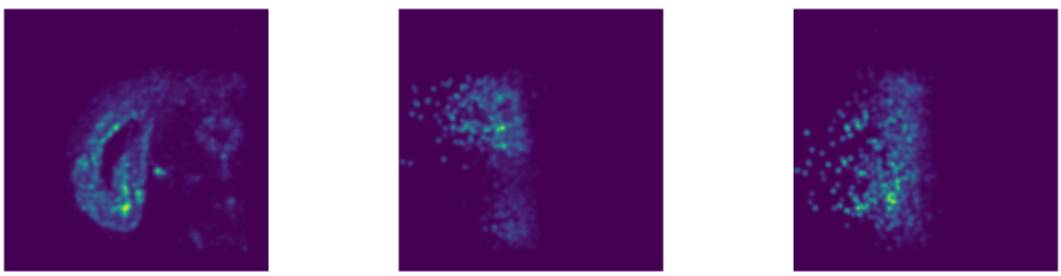
Real GEX (Tissue 1) - X-Y-Z-axis Tmsb10



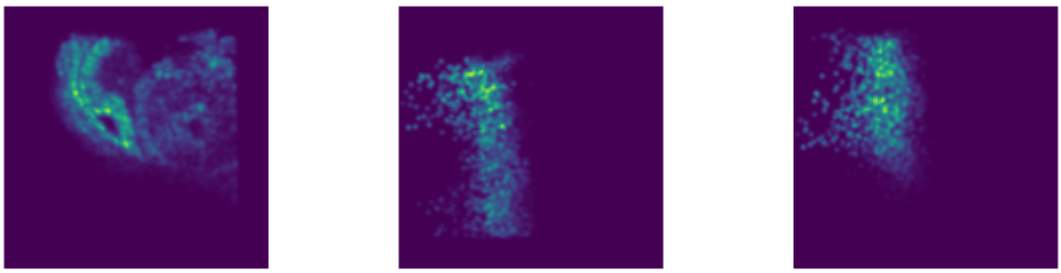
Real GEX (Tissue 2) - X-Y-Z-axis Tmsb10



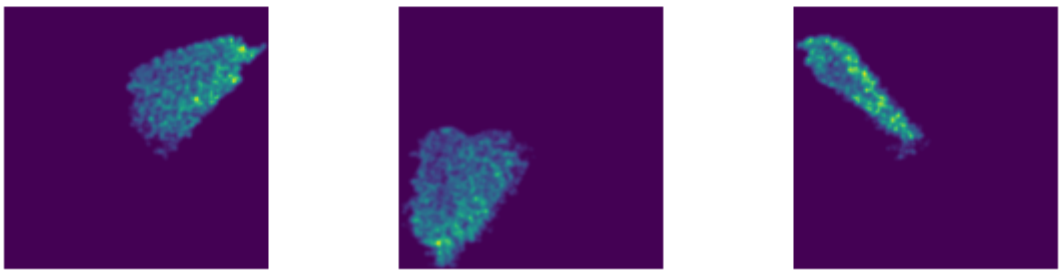
Real GEX (Tissue 3) - X-Y-Z-axis Tmsb10



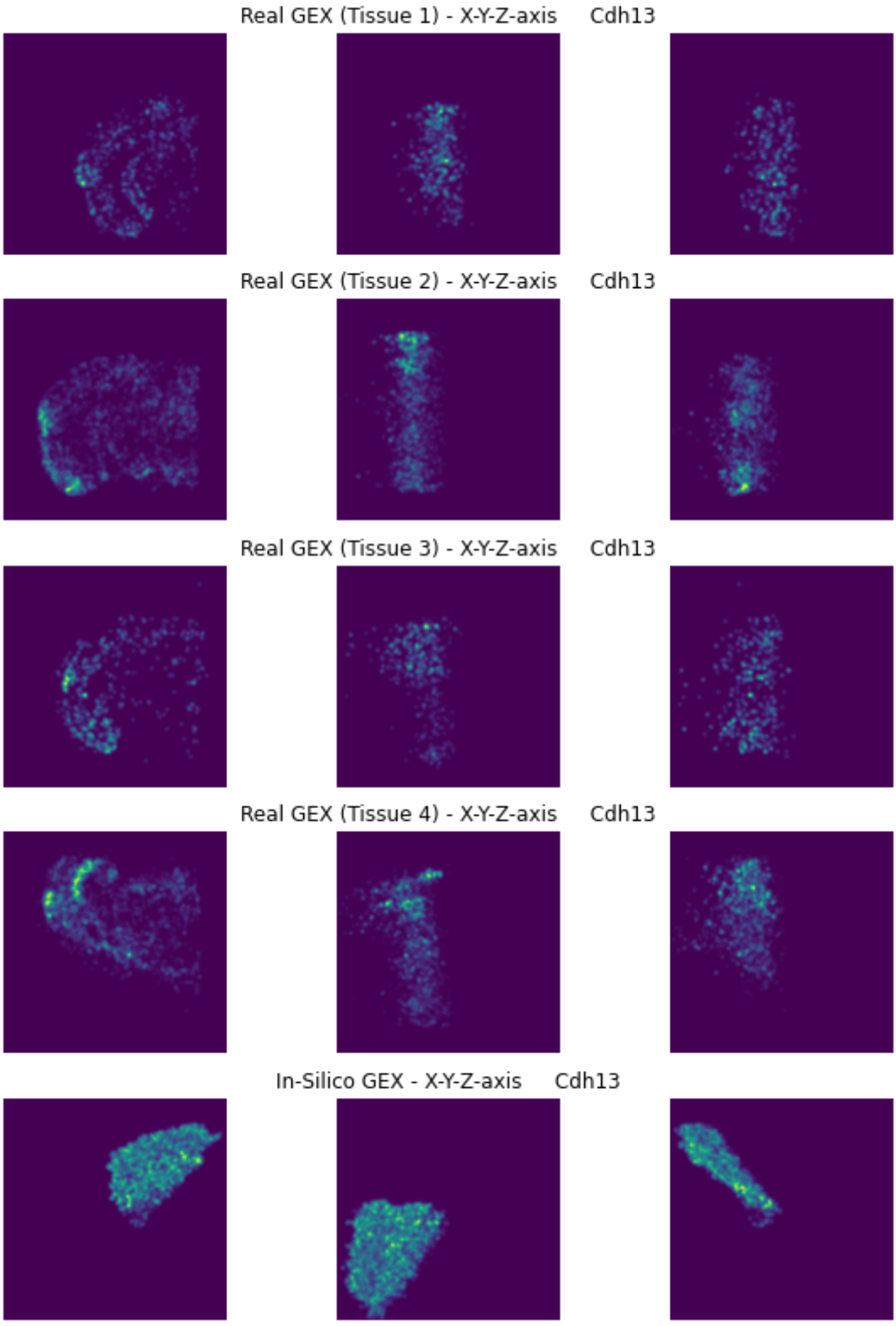
Real GEX (Tissue 4) - X-Y-Z-axis Tmsb10



In-Silico GEX - X-Y-Z-axis Tmsb10

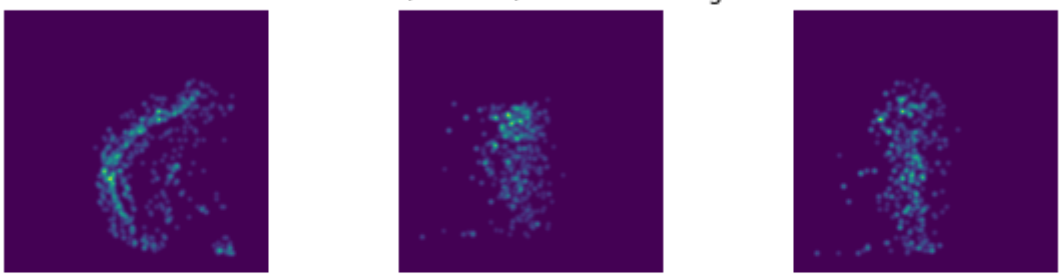


Real GEX Cdh13

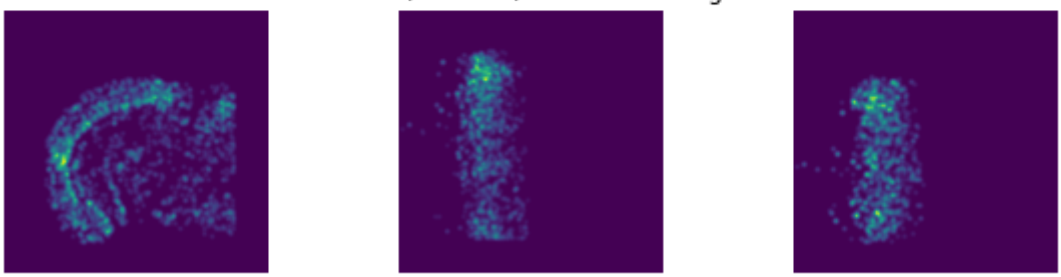


Real GEX Igsf21

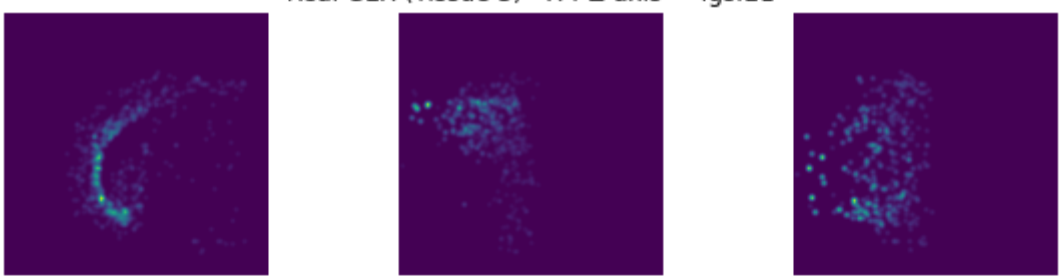
Real GEX (Tissue 1) - X-Y-Z-axis Igsf21



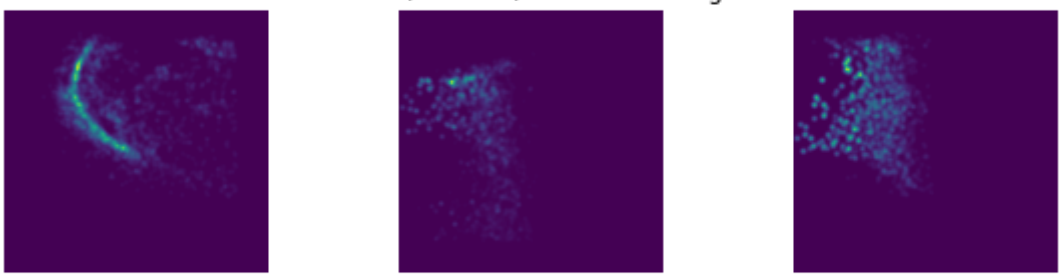
Real GEX (Tissue 2) - X-Y-Z-axis Igsf21



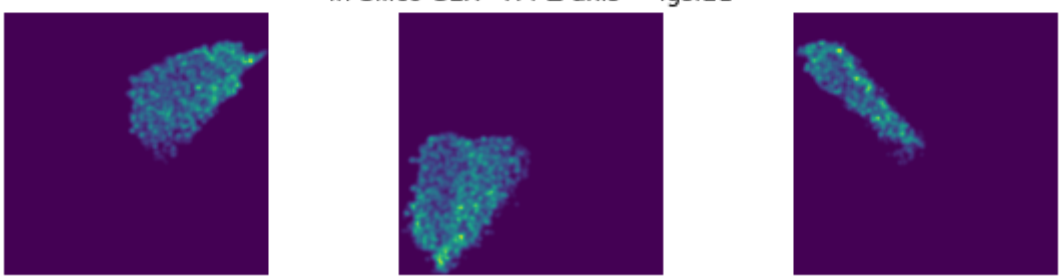
Real GEX (Tissue 3) - X-Y-Z-axis Igsf21



Real GEX (Tissue 4) - X-Y-Z-axis Igsf21

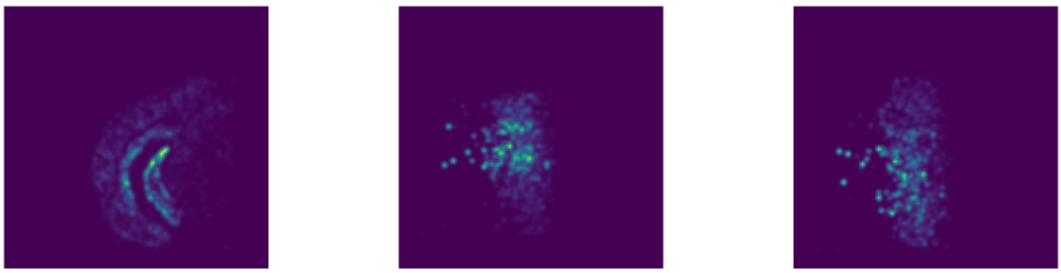


In-Silico GEX - X-Y-Z-axis Igsf21

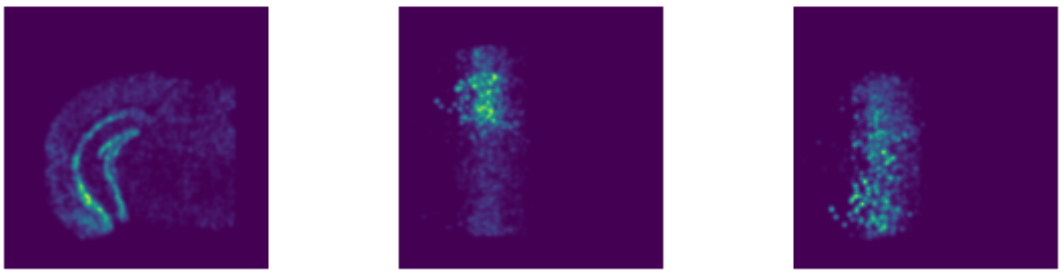


Real GEX Gria1

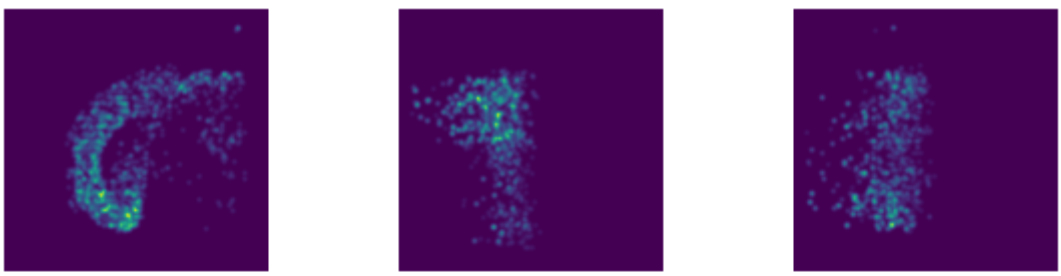
Real GEX (Tissue 1) - X-Y-Z-axis Gria1



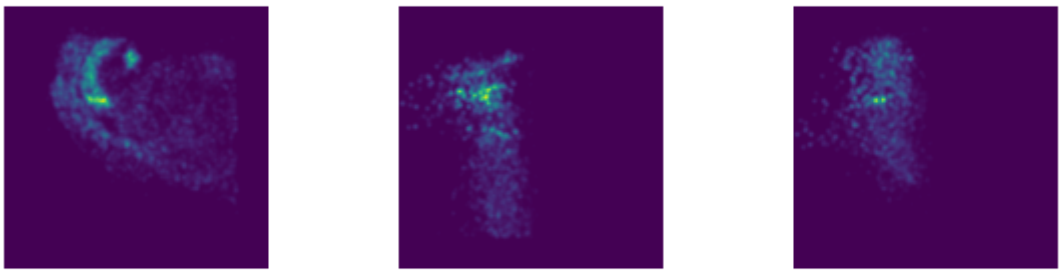
Real GEX (Tissue 2) - X-Y-Z-axis Gria1



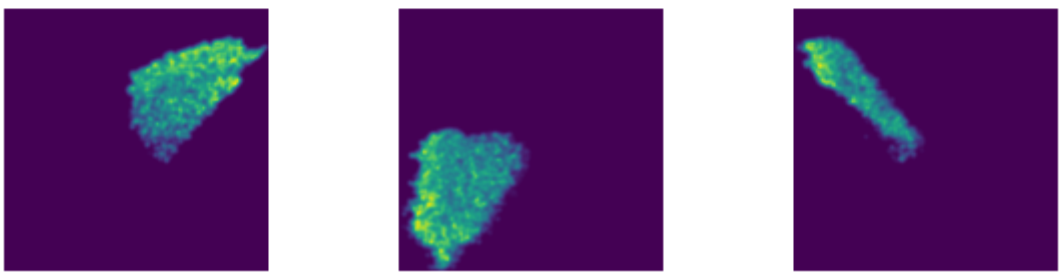
Real GEX (Tissue 3) - X-Y-Z-axis Gria1



Real GEX (Tissue 4) - X-Y-Z-axis Gria1

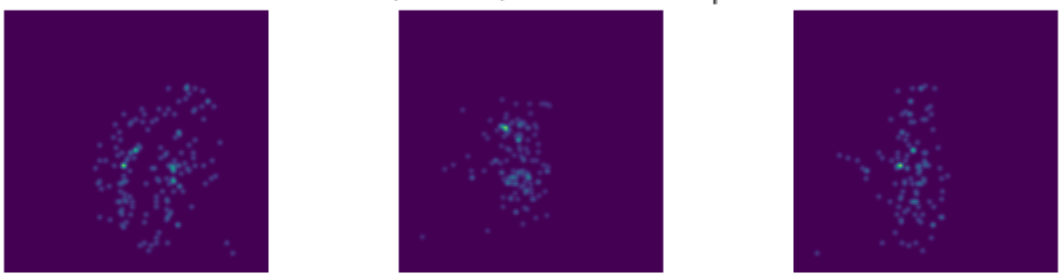


In-Silico GEX - X-Y-Z-axis Gria1

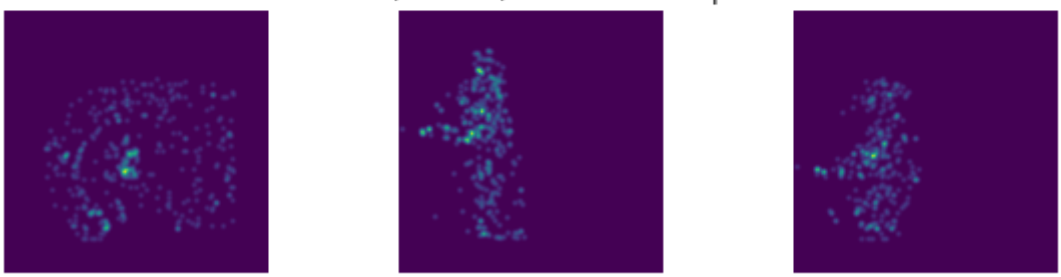


Real GEX Zfp804a

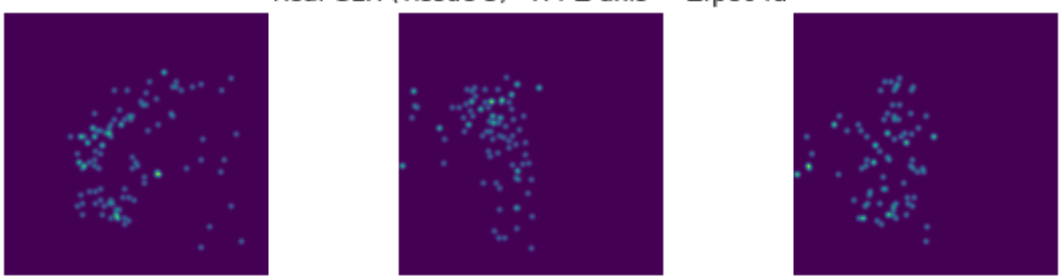
Real GEX (Tissue 1) - X-Y-Z-axis Zfp804a



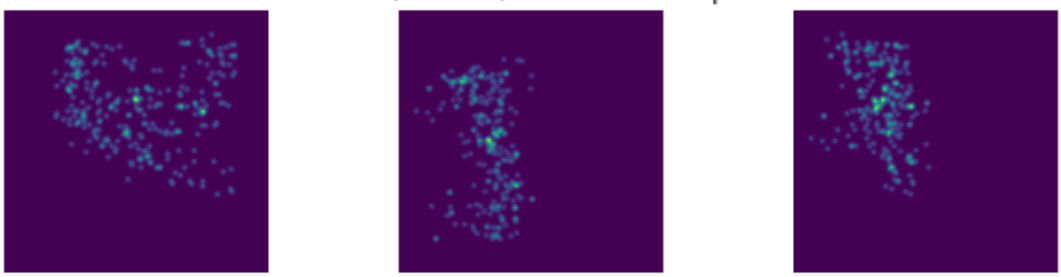
Real GEX (Tissue 2) - X-Y-Z-axis Zfp804a



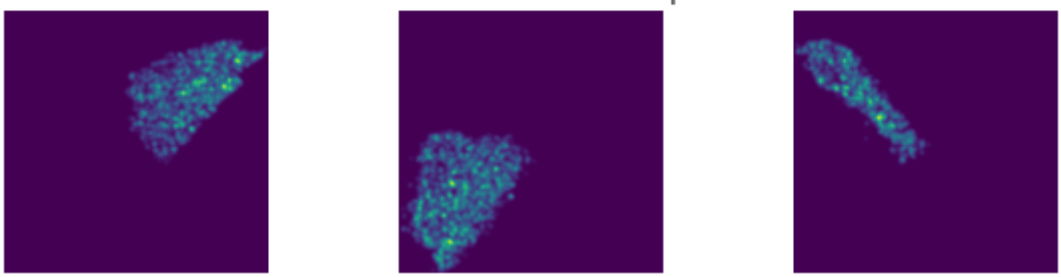
Real GEX (Tissue 3) - X-Y-Z-axis Zfp804a



Real GEX (Tissue 4) - X-Y-Z-axis Zfp804a

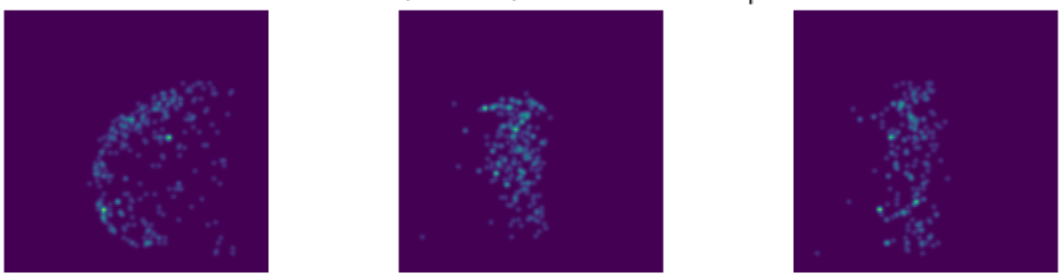


In-Silico GEX - X-Y-Z-axis Zfp804a

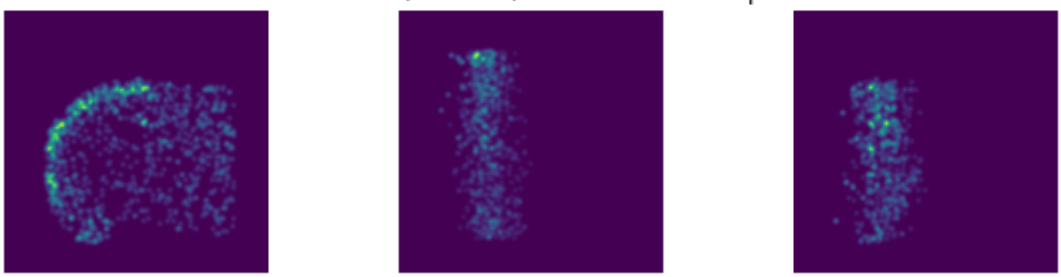


Real GEX Brinp3

Real GEX (Tissue 1) - X-Y-Z-axis Brinp3

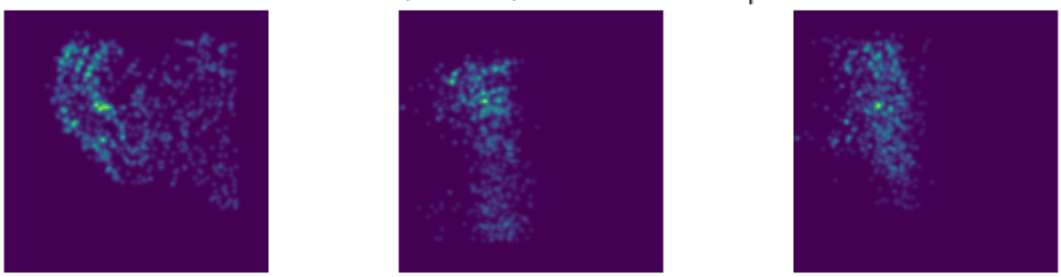


Real GEX (Tissue 2) - X-Y-Z-axis Brinp3

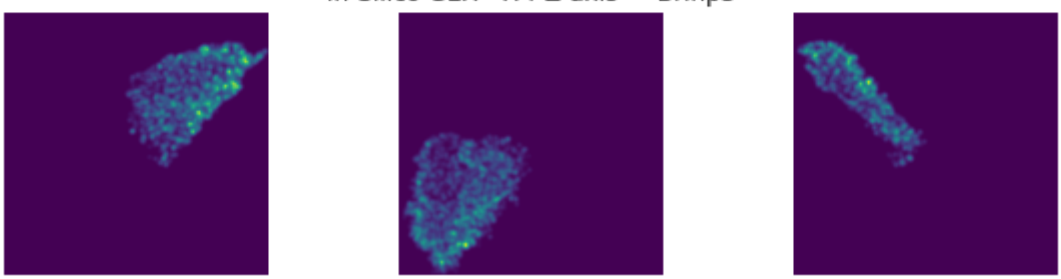


Real GEX (Tissue 3) - X-Y-Z-axis Brinp3

Real GEX (Tissue 4) - X-Y-Z-axis Brinp3

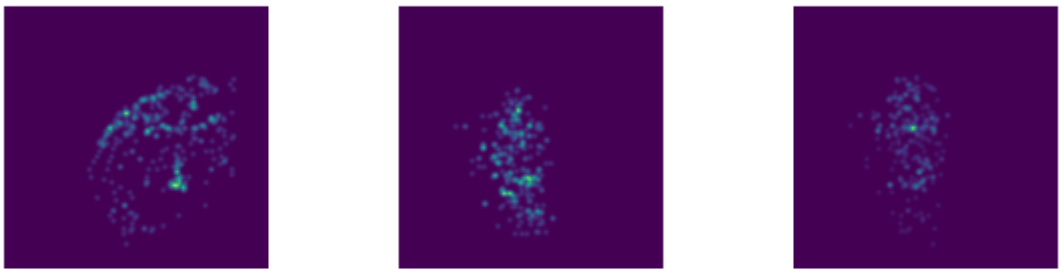


In-Silico GEX - X-Y-Z-axis Brinp3

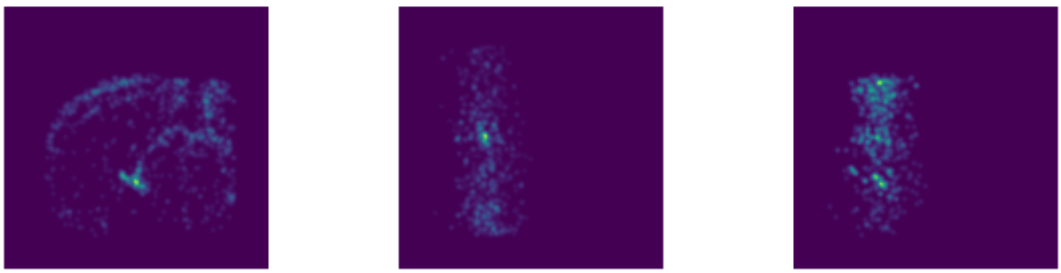


Real GEX Cbln2

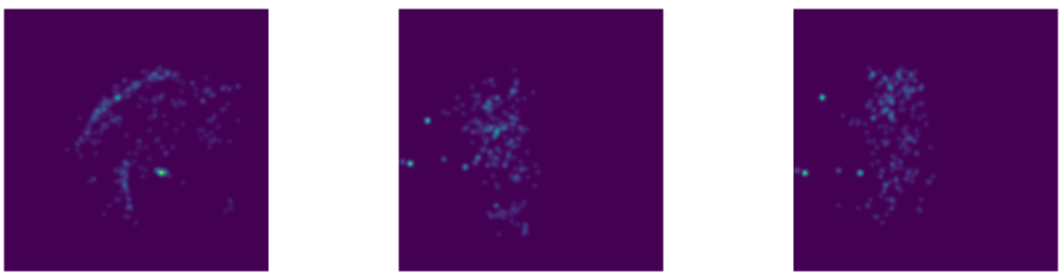
Real GEX (Tissue 1) - X-Y-Z-axis Cbln2



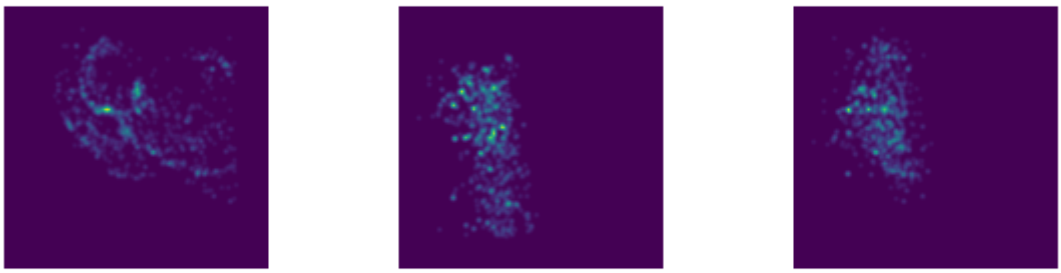
Real GEX (Tissue 2) - X-Y-Z-axis Cbln2



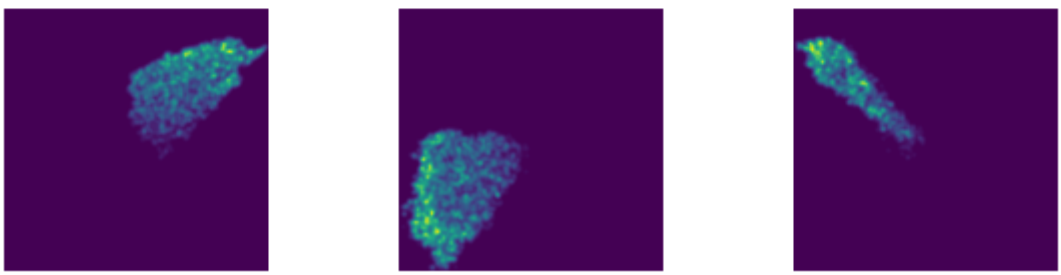
Real GEX (Tissue 3) - X-Y-Z-axis Cbln2



Real GEX (Tissue 4) - X-Y-Z-axis Cbln2

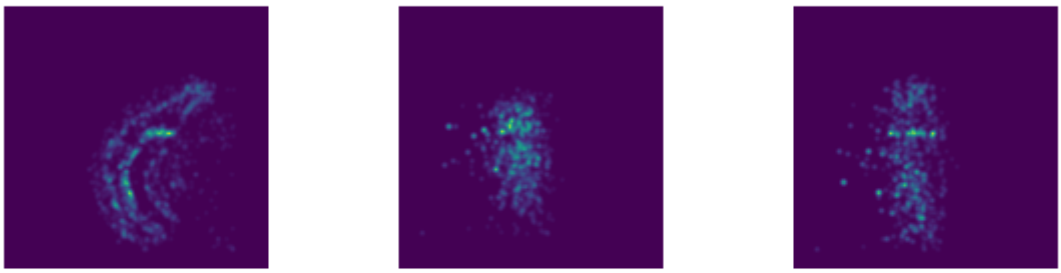


In-Silico GEX - X-Y-Z-axis Cbln2

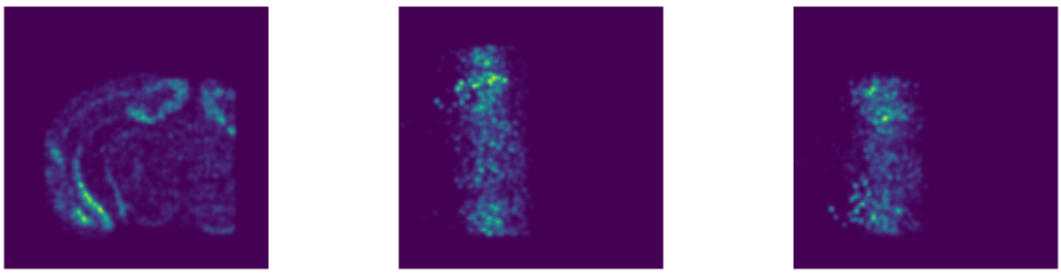


Real GEX Etv1

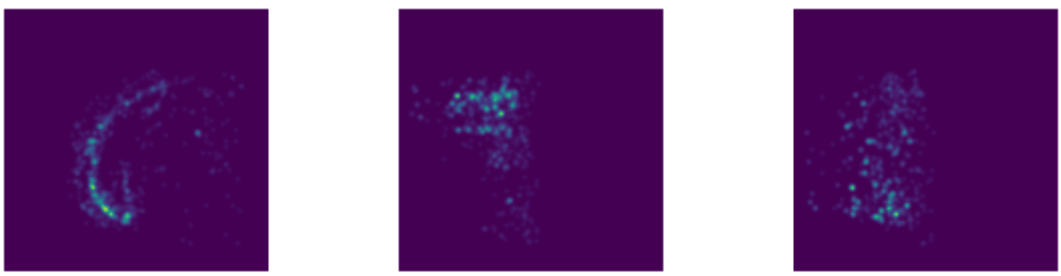
Real GEX (Tissue 1) - X-Y-Z-axis Etv1



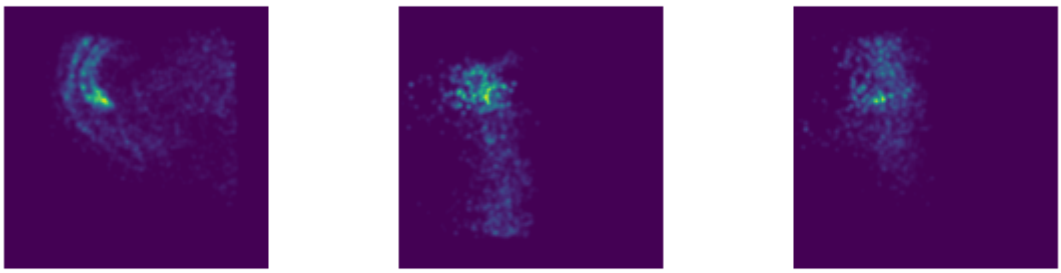
Real GEX (Tissue 2) - X-Y-Z-axis Etv1



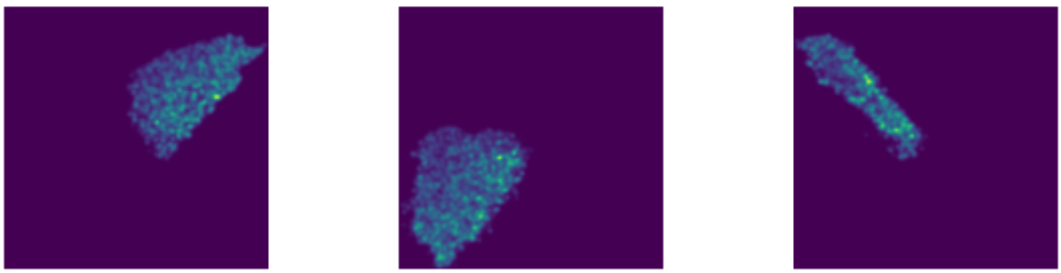
Real GEX (Tissue 3) - X-Y-Z-axis Etv1



Real GEX (Tissue 4) - X-Y-Z-axis Etv1

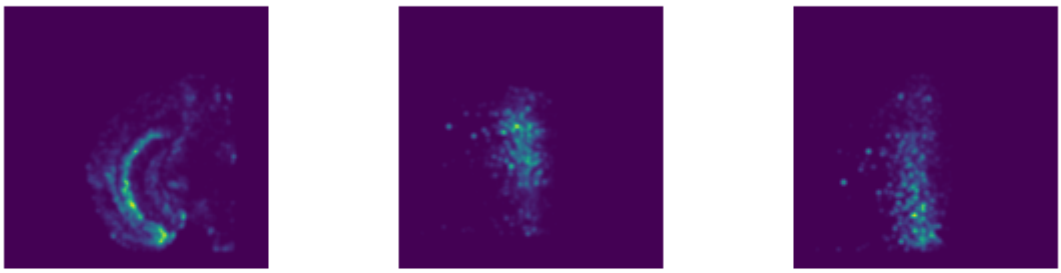


In-Silico GEX - X-Y-Z-axis Etv1

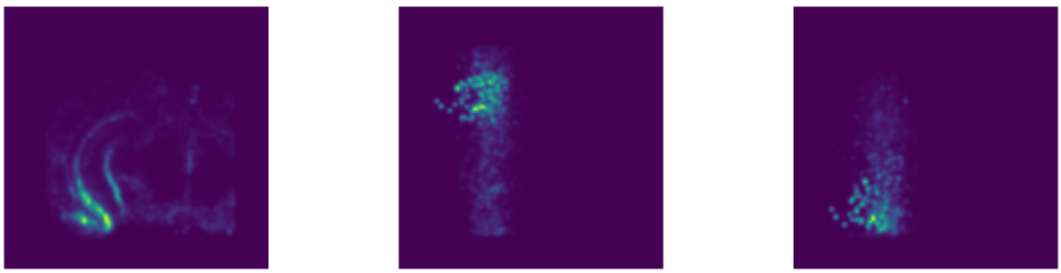


Real GEX Nnat

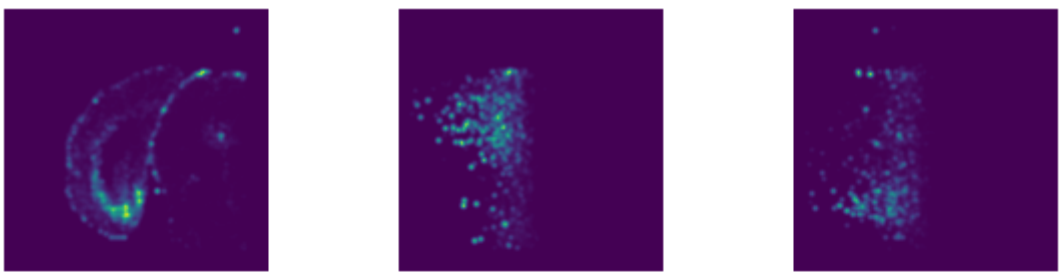
Real GEX (Tissue 1) - X-Y-Z-axis Nnat



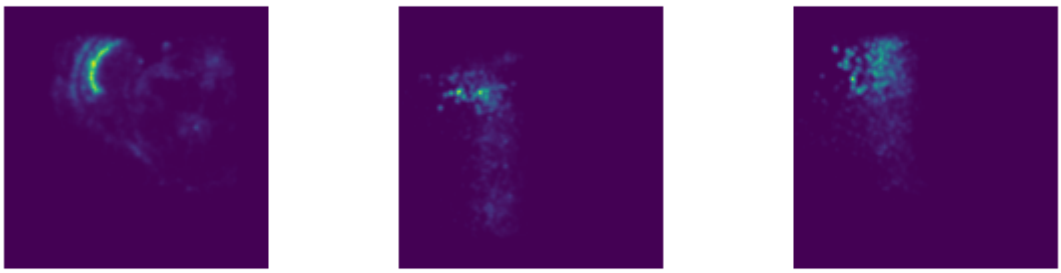
Real GEX (Tissue 2) - X-Y-Z-axis Nnat



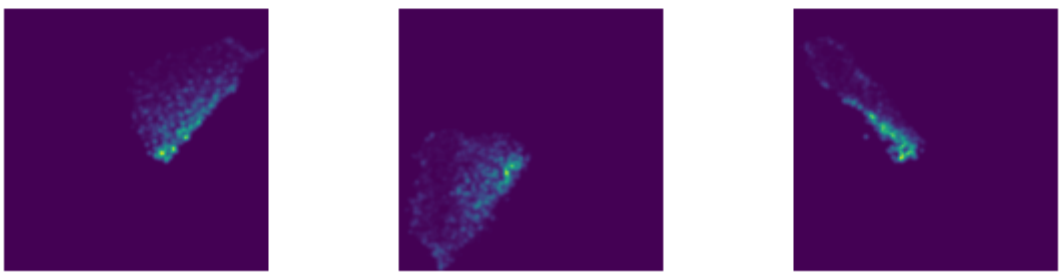
Real GEX (Tissue 3) - X-Y-Z-axis Nnat



Real GEX (Tissue 4) - X-Y-Z-axis Nnat

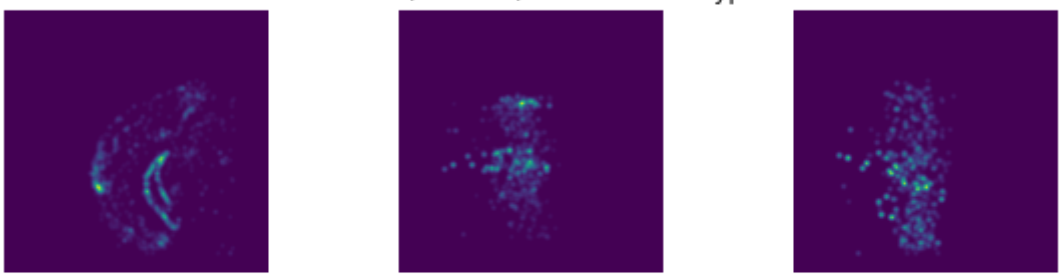


In-Silico GEX - X-Y-Z-axis Nnat

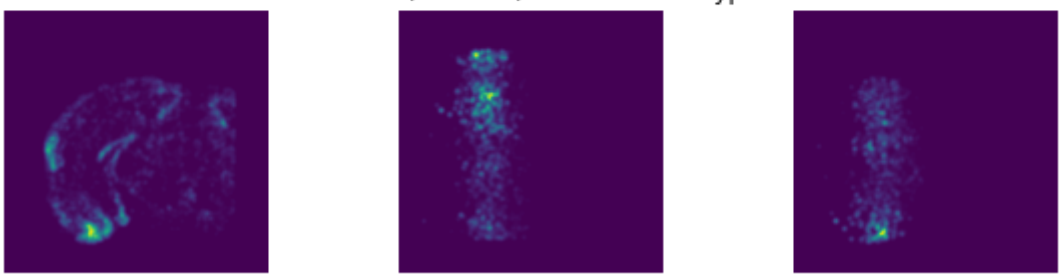


Real GEX Lypd1

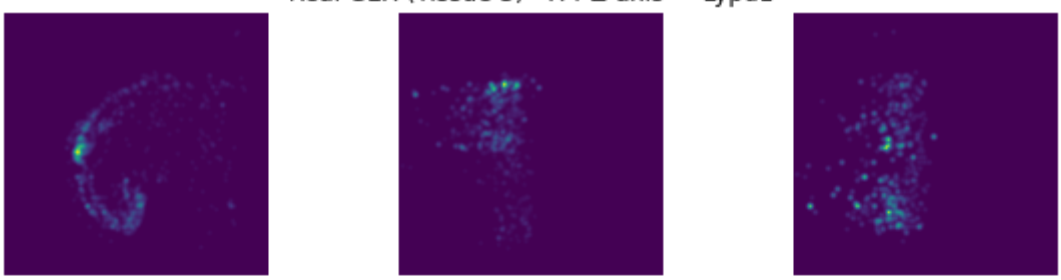
Real GEX (Tissue 1) - X-Y-Z-axis Lypd1



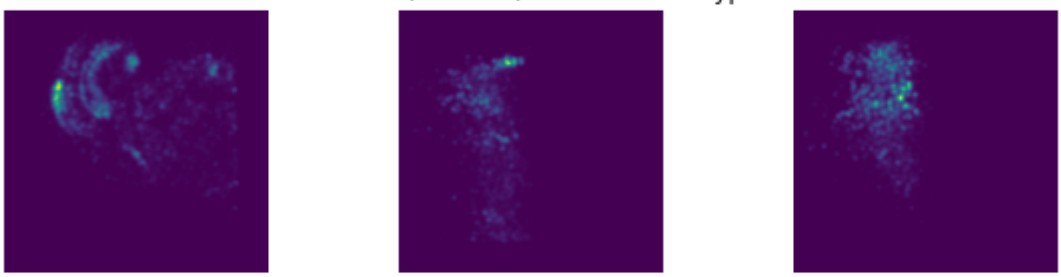
Real GEX (Tissue 2) - X-Y-Z-axis Lypd1



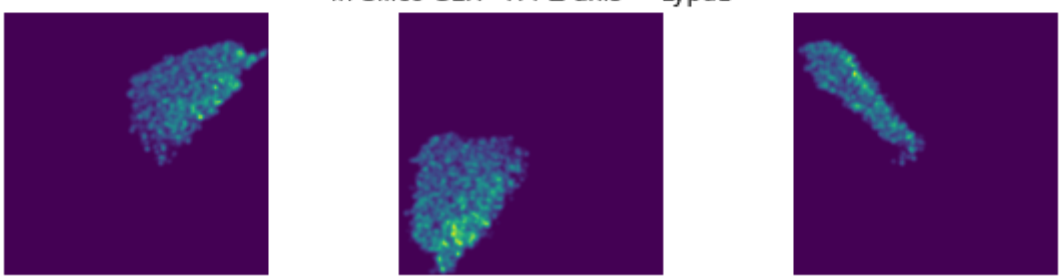
Real GEX (Tissue 3) - X-Y-Z-axis Lypd1



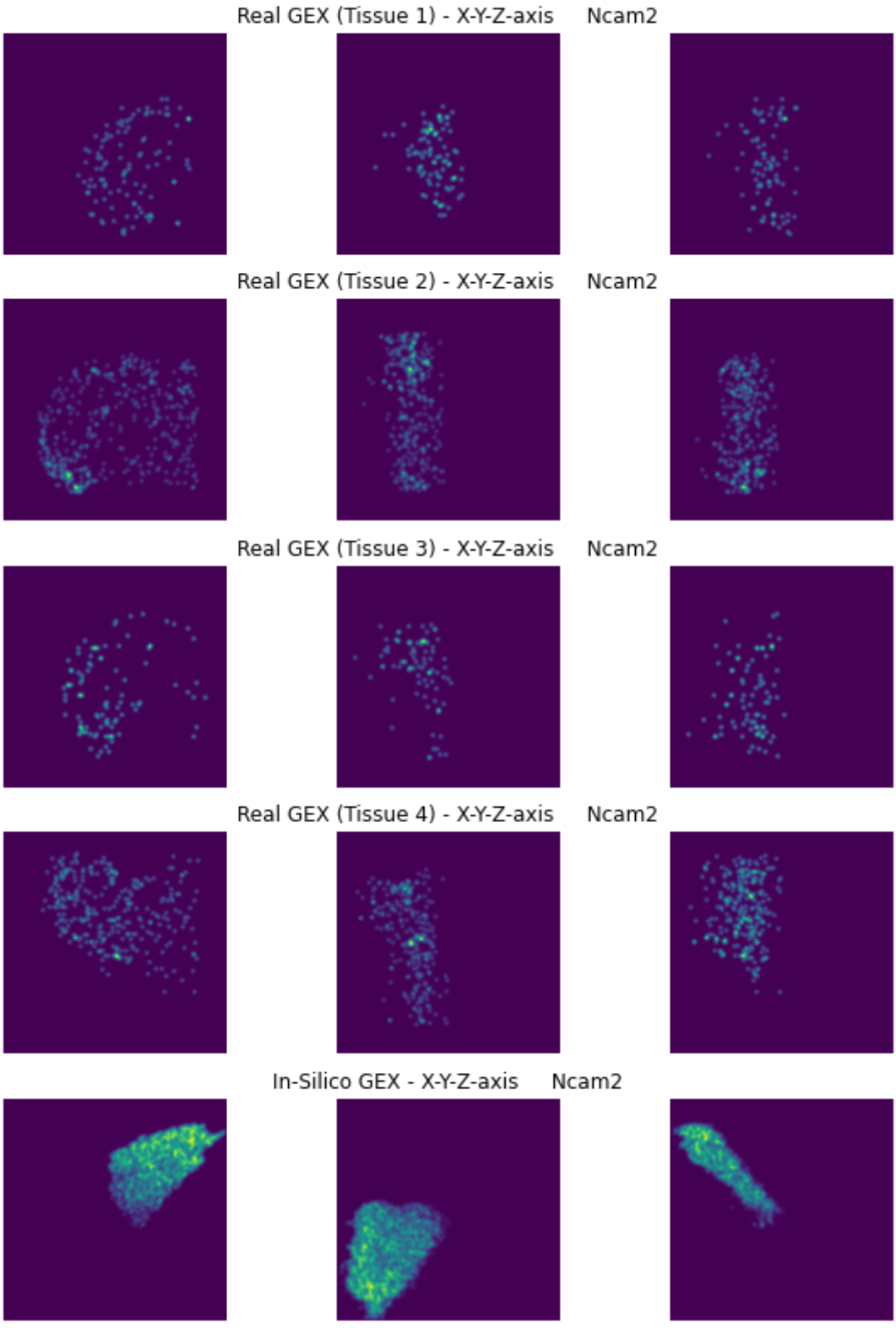
Real GEX (Tissue 4) - X-Y-Z-axis Lypd1



In-Silico GEX - X-Y-Z-axis Lypd1

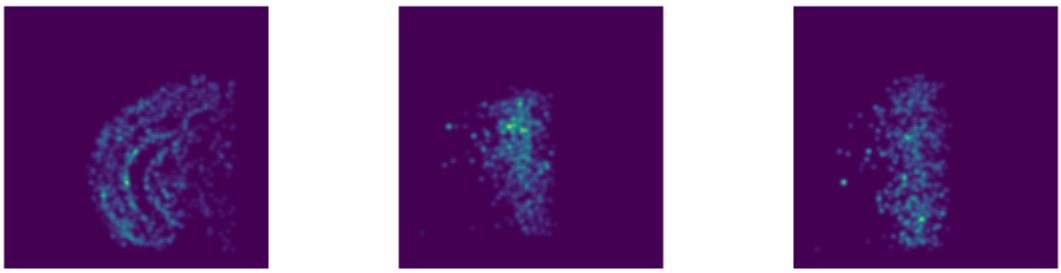


Real GEX Ncam2

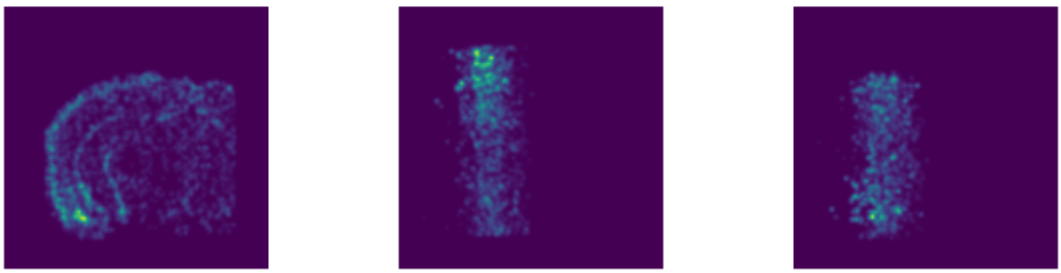


Real GEX Alcam

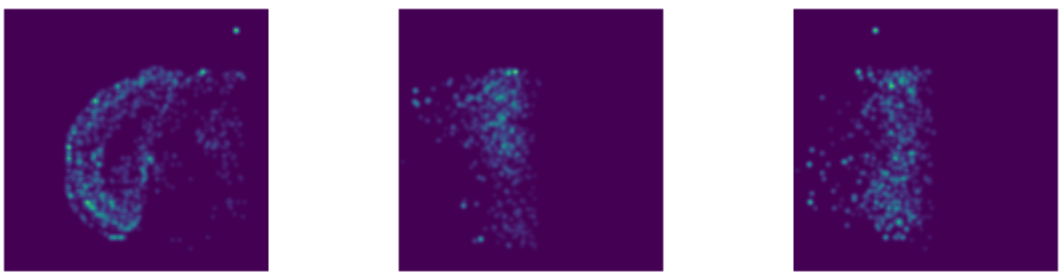
Real GEX (Tissue 1) - X-Y-Z-axis Alcam



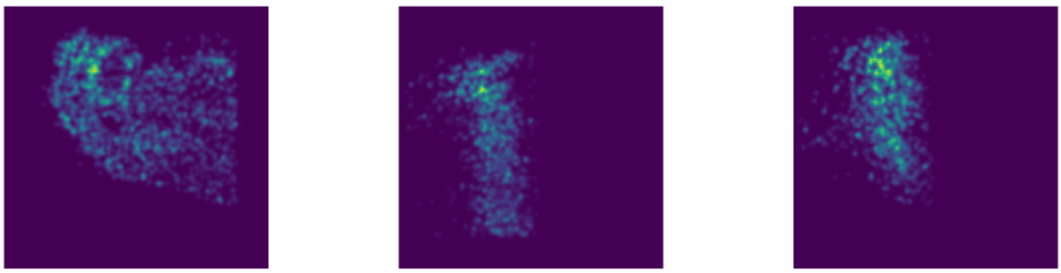
Real GEX (Tissue 2) - X-Y-Z-axis Alcam



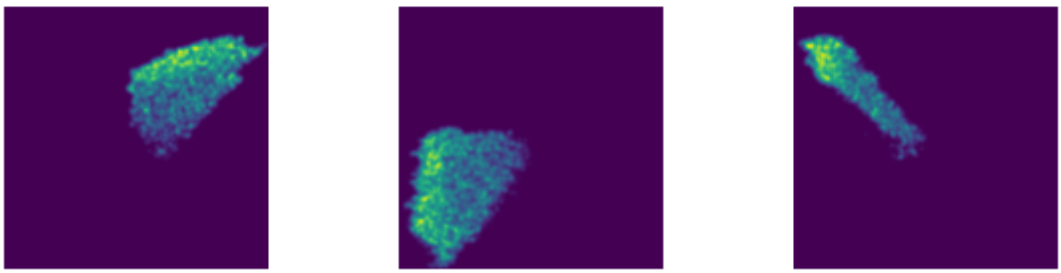
Real GEX (Tissue 3) - X-Y-Z-axis Alcam



Real GEX (Tissue 4) - X-Y-Z-axis Alcam

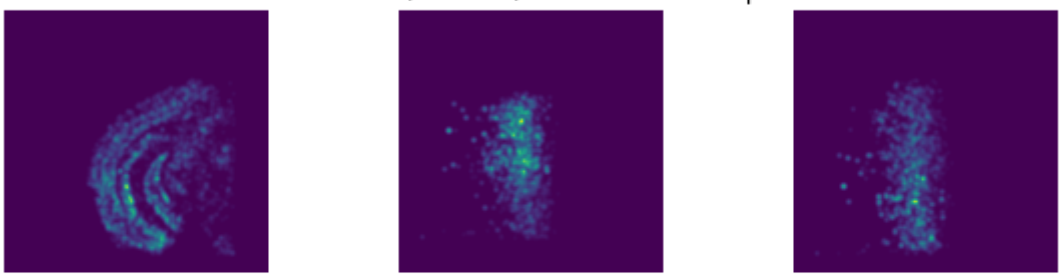


In-Silico GEX - X-Y-Z-axis Alcam

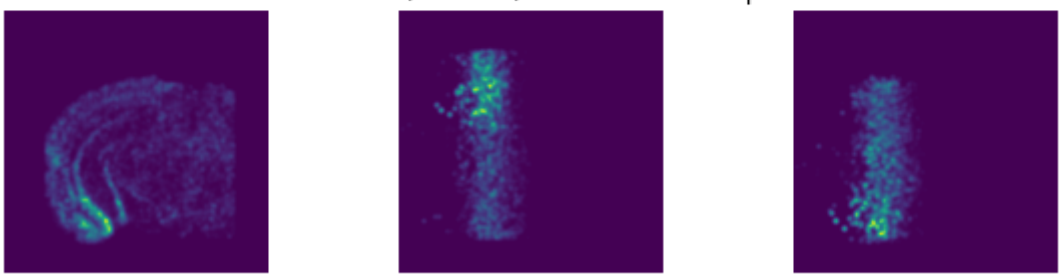


Real GEX Timp2

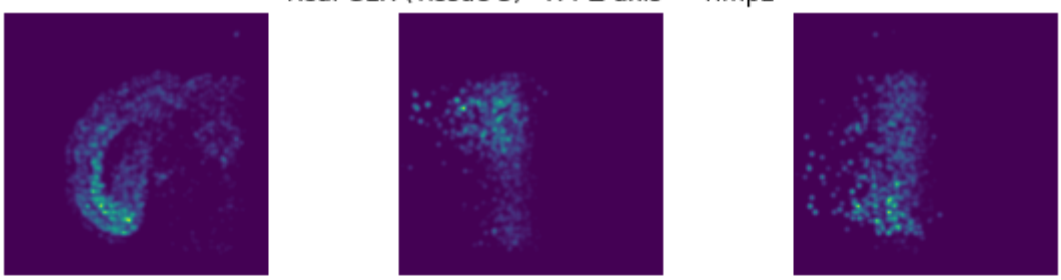
Real GEX (Tissue 1) - X-Y-Z-axis Timp2



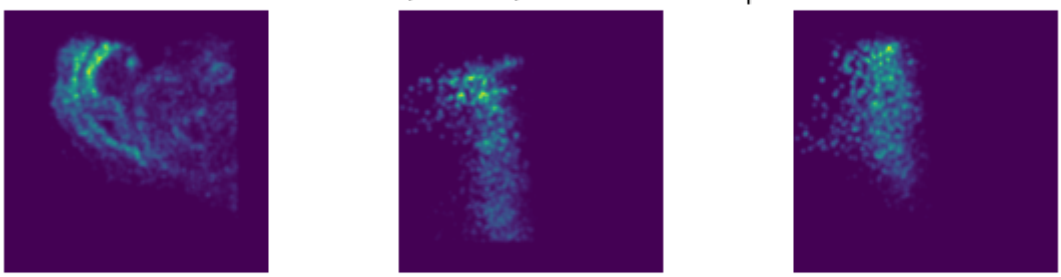
Real GEX (Tissue 2) - X-Y-Z-axis Timp2



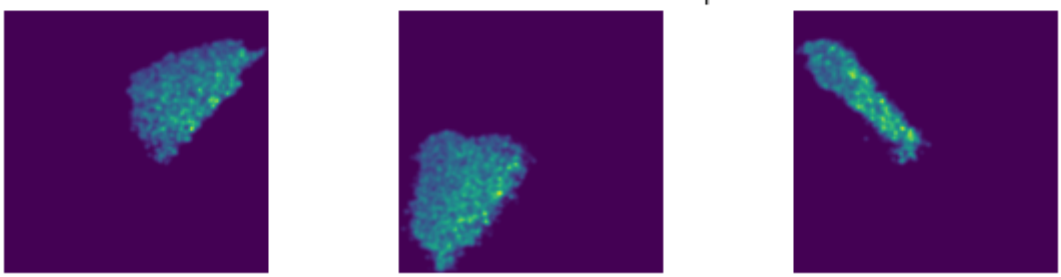
Real GEX (Tissue 3) - X-Y-Z-axis Timp2



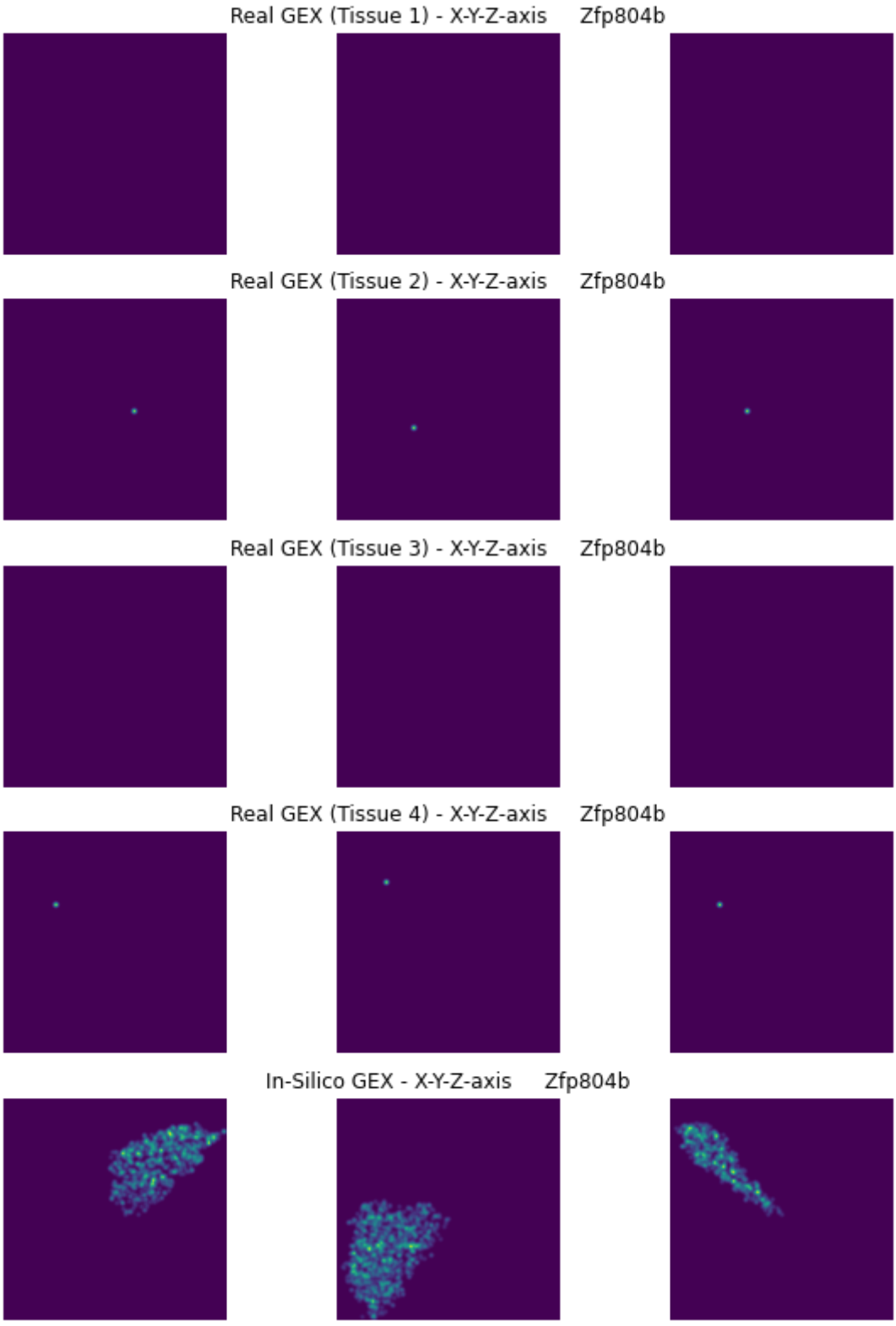
Real GEX (Tissue 4) - X-Y-Z-axis Timp2



In-Silico GEX - X-Y-Z-axis Timp2

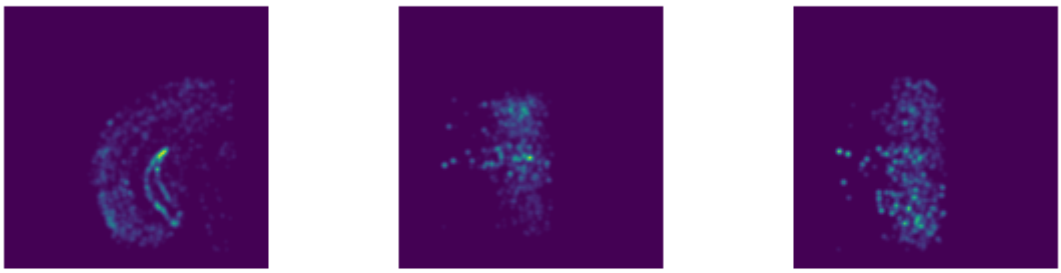


Real GEX Zfp804b

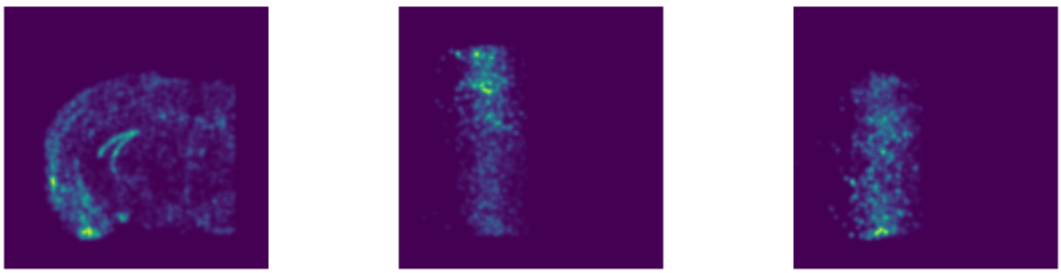


Real GEX Marcksl1

Real GEX (Tissue 1) - X-Y-Z-axis Marcksl1

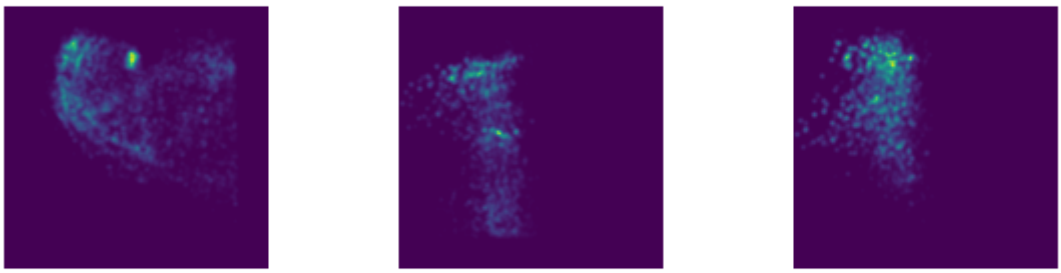


Real GEX (Tissue 2) - X-Y-Z-axis Marcksl1



Real GEX (Tissue 3) - X-Y-Z-axis Marcksl1

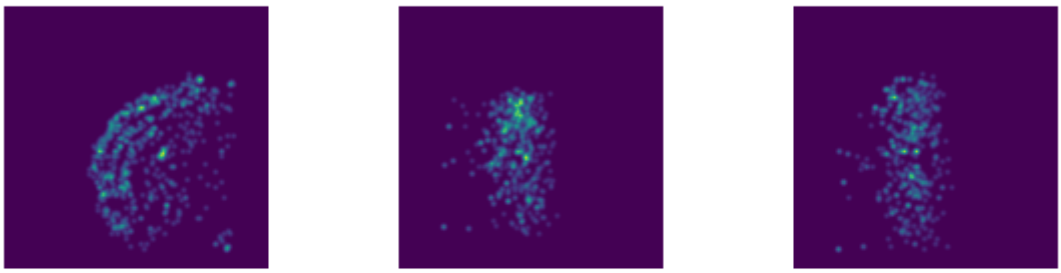
Real GEX (Tissue 4) - X-Y-Z-axis Marcksl1



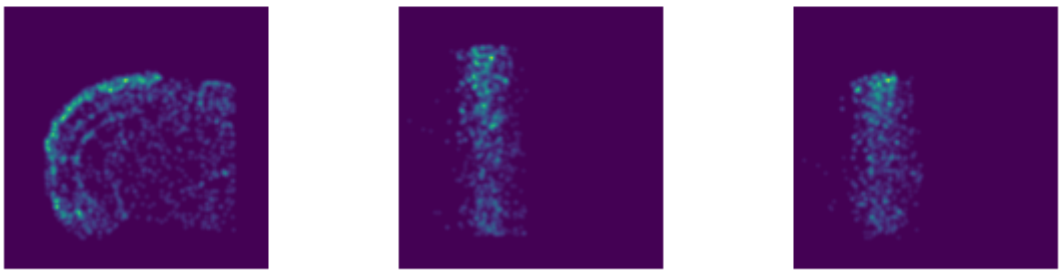
In-Silico GEX - X-Y-Z-axis Marcksl1

Real GEX Kctd1

Real GEX (Tissue 1) - X-Y-Z-axis Kctd1

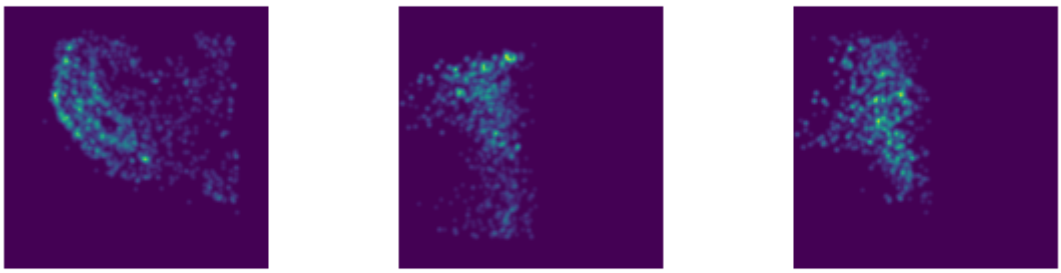


Real GEX (Tissue 2) - X-Y-Z-axis Kctd1

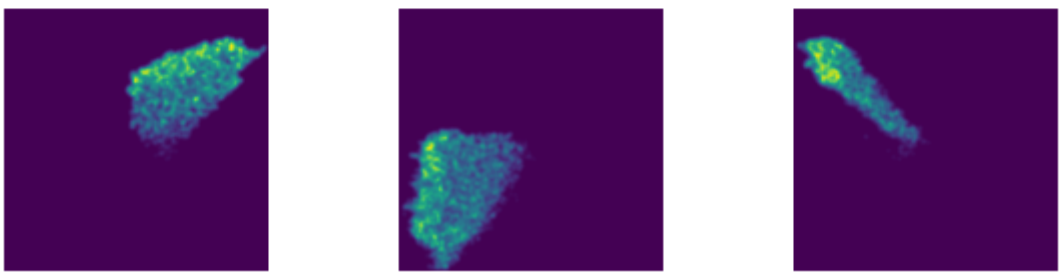


Real GEX (Tissue 3) - X-Y-Z-axis Kctd1

Real GEX (Tissue 4) - X-Y-Z-axis Kctd1

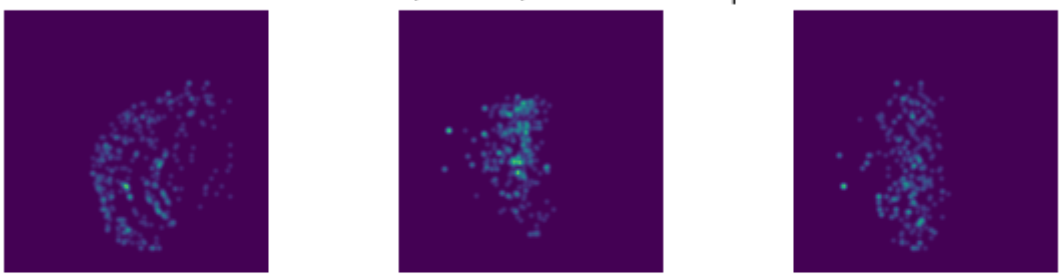


In-Silico GEX - X-Y-Z-axis Kctd1

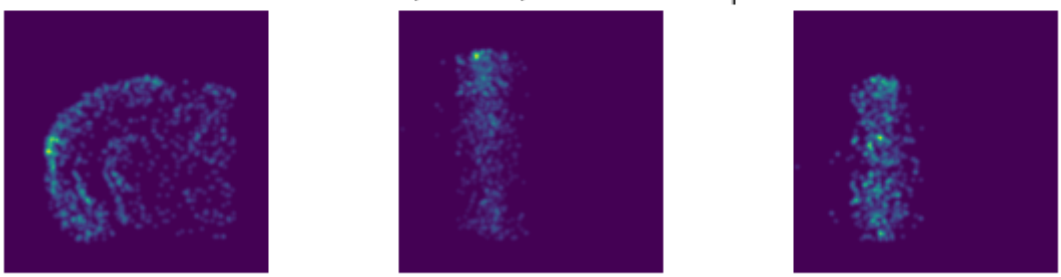


Real GEX Ptprk

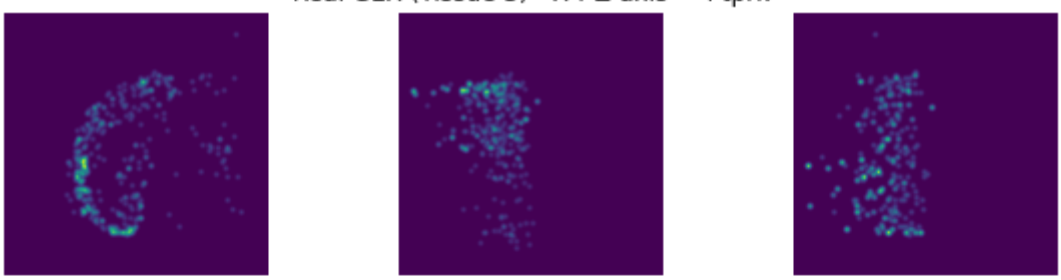
Real GEX (Tissue 1) - X-Y-Z-axis Ptprk



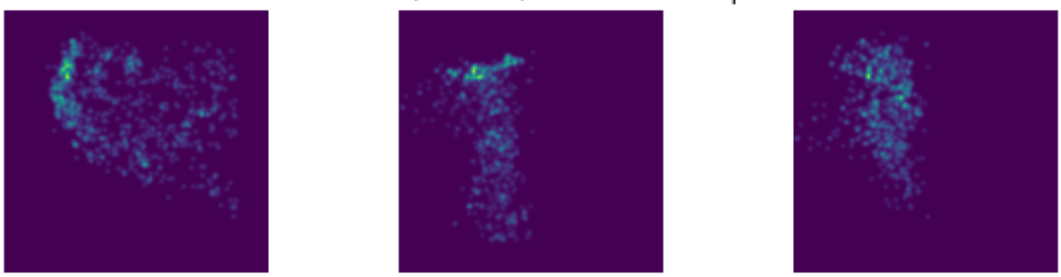
Real GEX (Tissue 2) - X-Y-Z-axis Ptprk



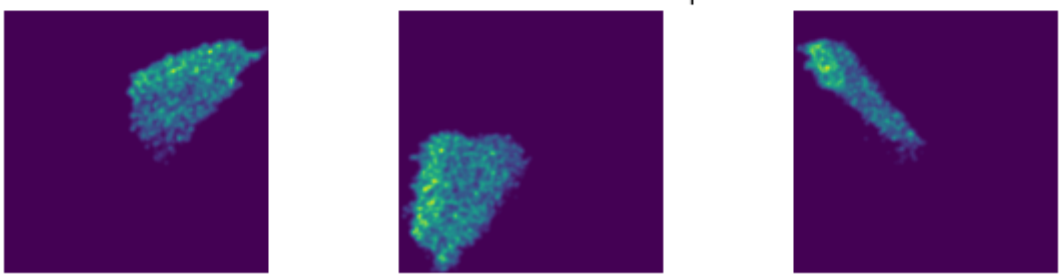
Real GEX (Tissue 3) - X-Y-Z-axis Ptprk



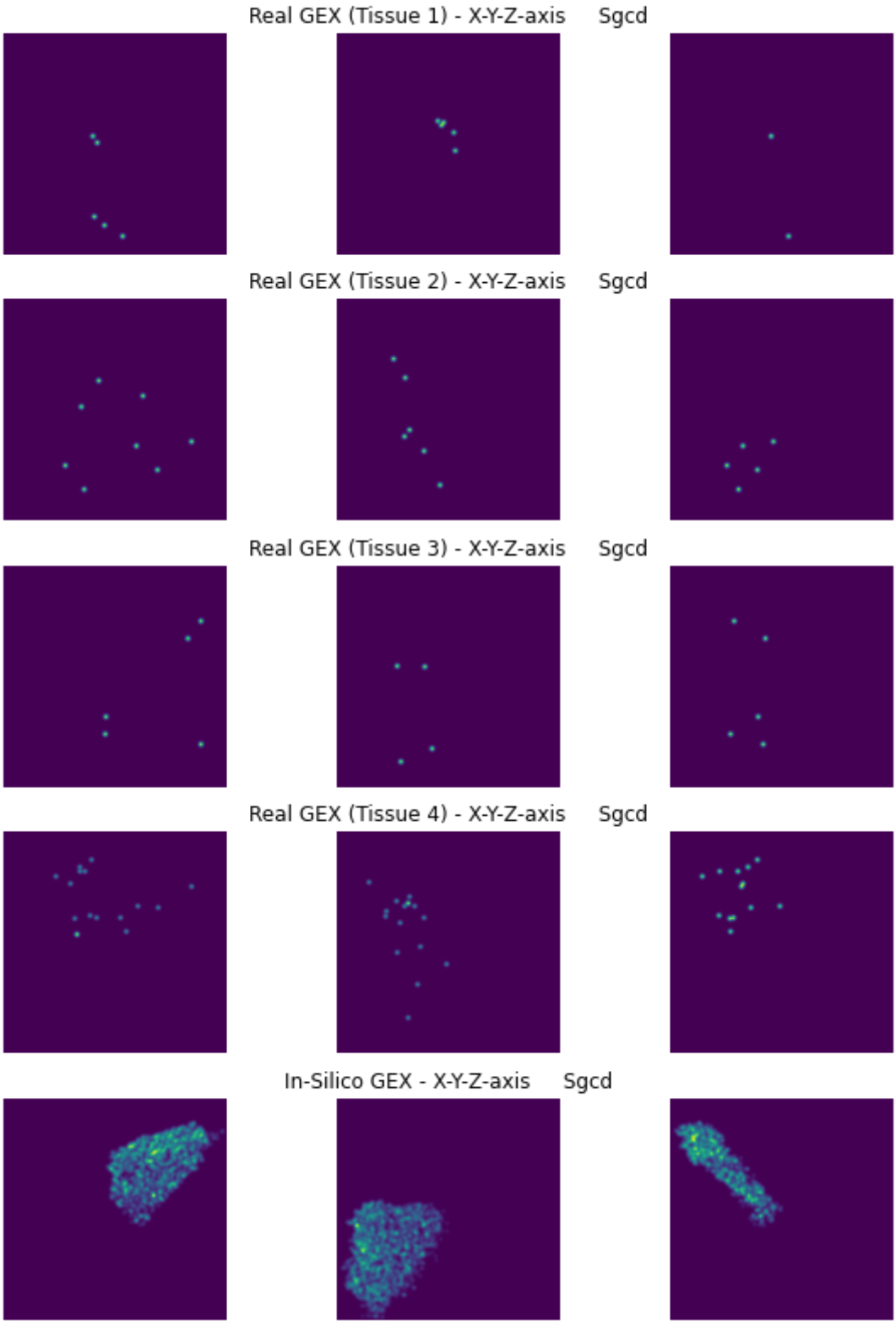
Real GEX (Tissue 4) - X-Y-Z-axis Ptprk



In-Silico GEX - X-Y-Z-axis Ptprk

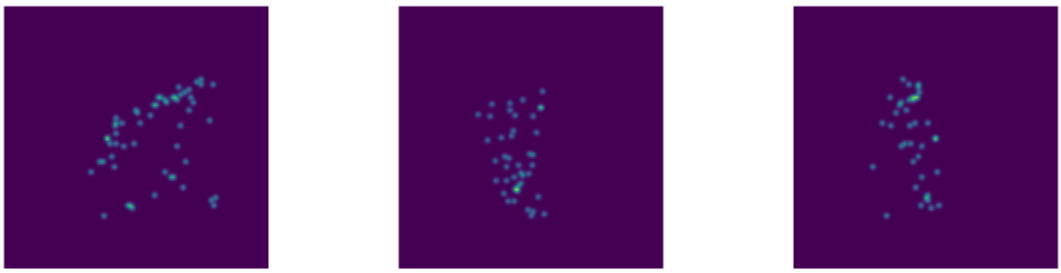


Real GEX Sgcd

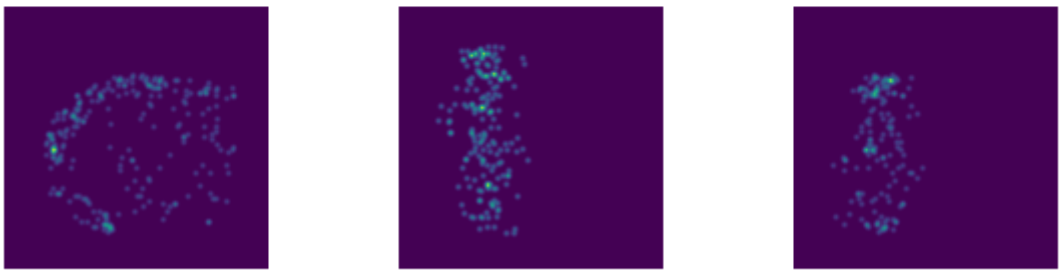


Real GEX Efna5

Real GEX (Tissue 1) - X-Y-Z-axis Efna5

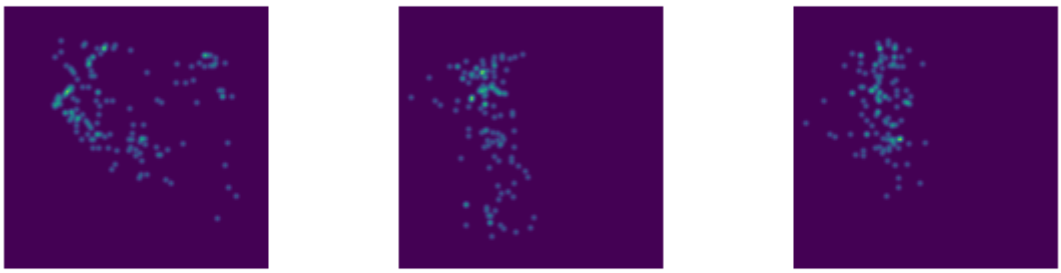


Real GEX (Tissue 2) - X-Y-Z-axis Efna5

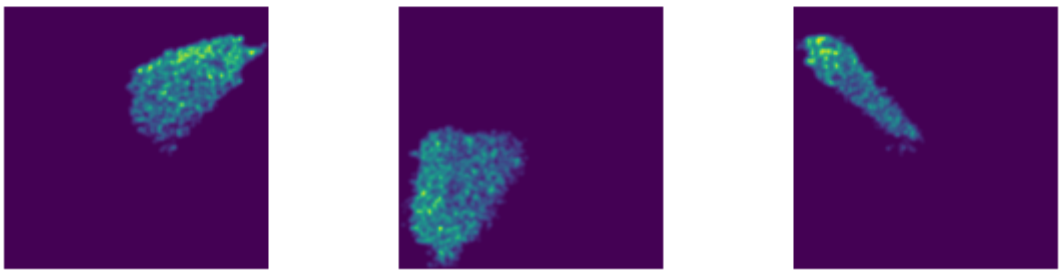


Real GEX (Tissue 3) - X-Y-Z-axis Efna5

Real GEX (Tissue 4) - X-Y-Z-axis Efna5

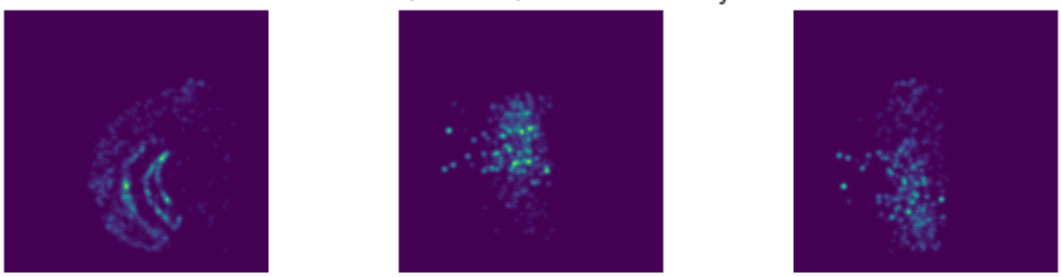


In-Silico GEX - X-Y-Z-axis Efna5

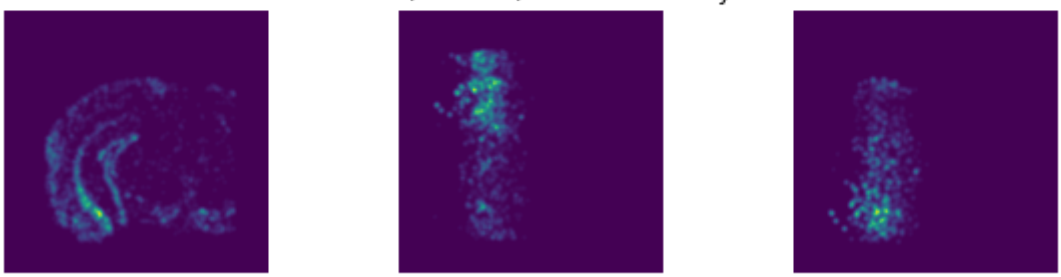


Real GEX Syt17

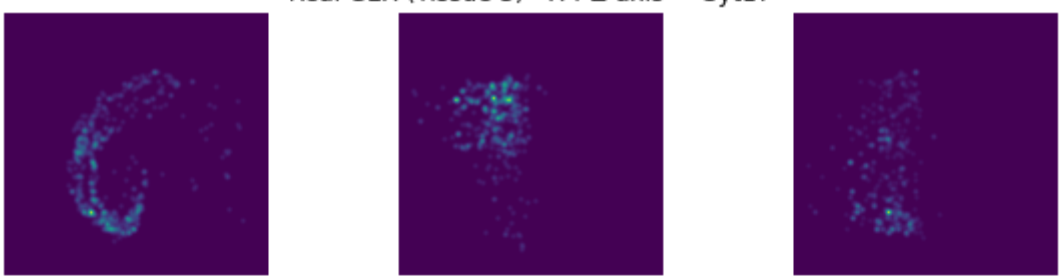
Real GEX (Tissue 1) - X-Y-Z-axis Syt17



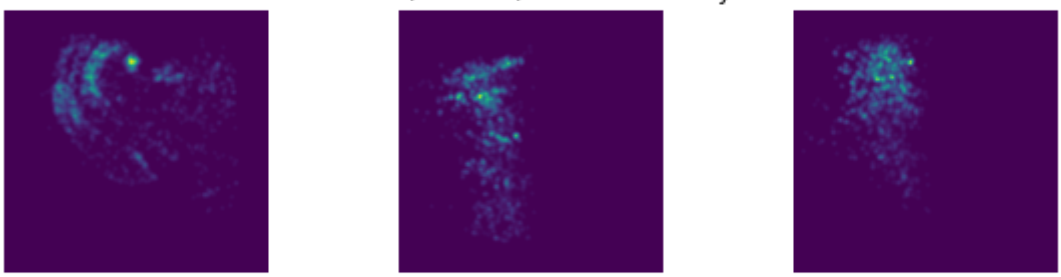
Real GEX (Tissue 2) - X-Y-Z-axis Syt17



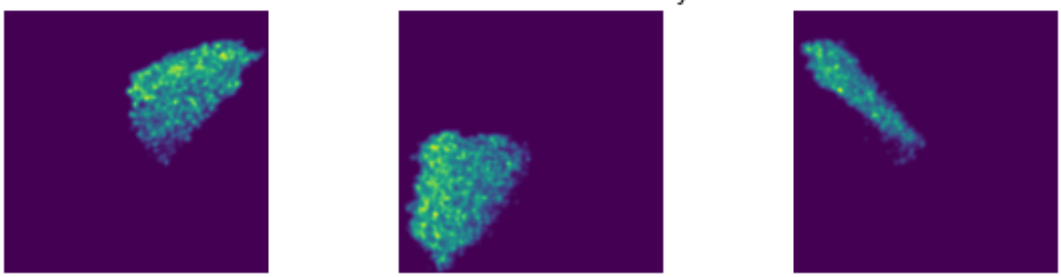
Real GEX (Tissue 3) - X-Y-Z-axis Syt17



Real GEX (Tissue 4) - X-Y-Z-axis Syt17

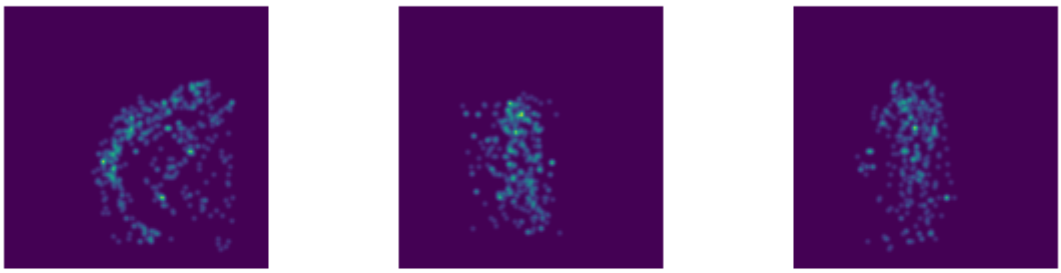


In-Silico GEX - X-Y-Z-axis Syt17

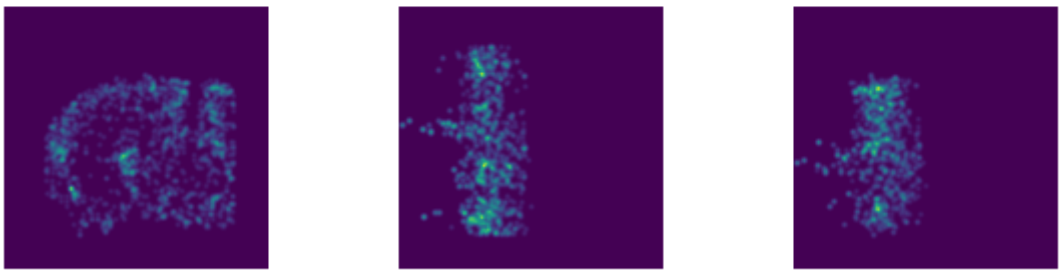


Real GEX Olfm3

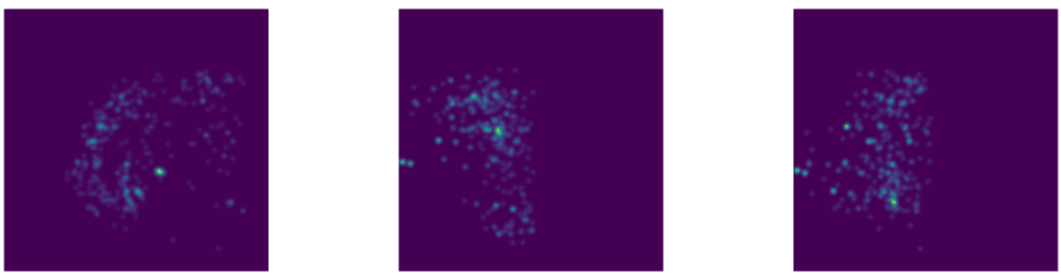
Real GEX (Tissue 1) - X-Y-Z-axis Olfm3



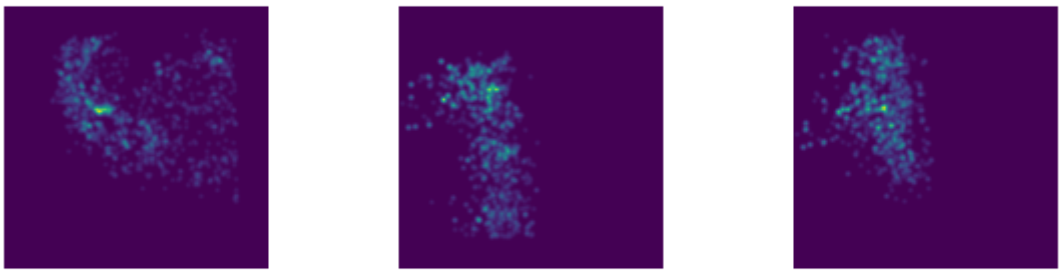
Real GEX (Tissue 2) - X-Y-Z-axis Olfm3



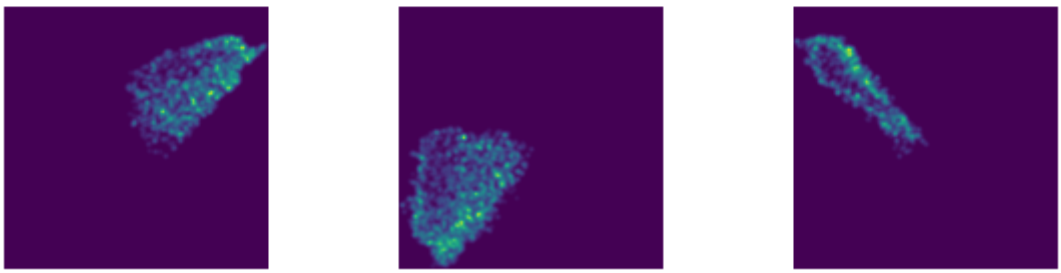
Real GEX (Tissue 3) - X-Y-Z-axis Olfm3



Real GEX (Tissue 4) - X-Y-Z-axis Olfm3

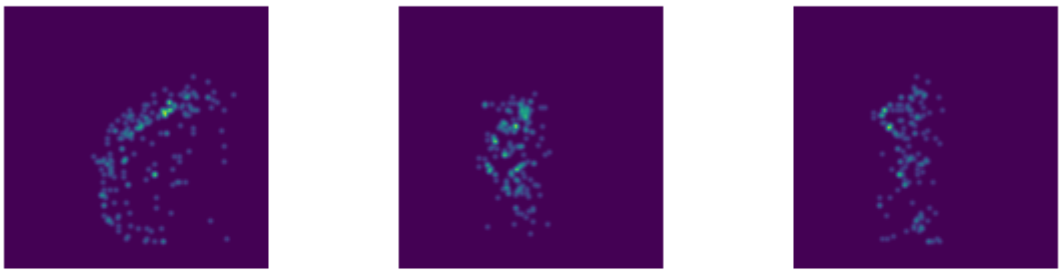


In-Silico GEX - X-Y-Z-axis Olfm3

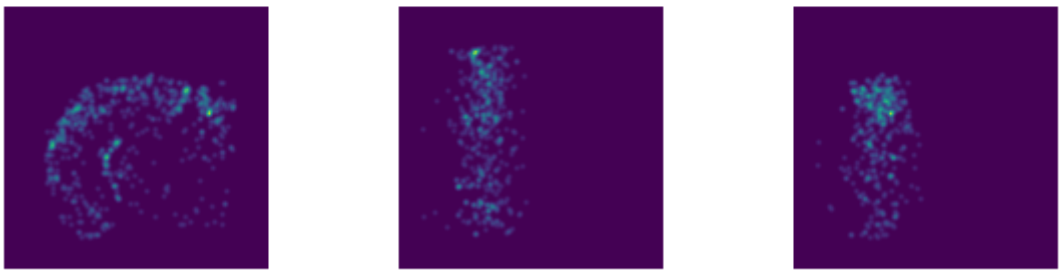


Real GEX Trhde

Real GEX (Tissue 1) - X-Y-Z-axis Trhde

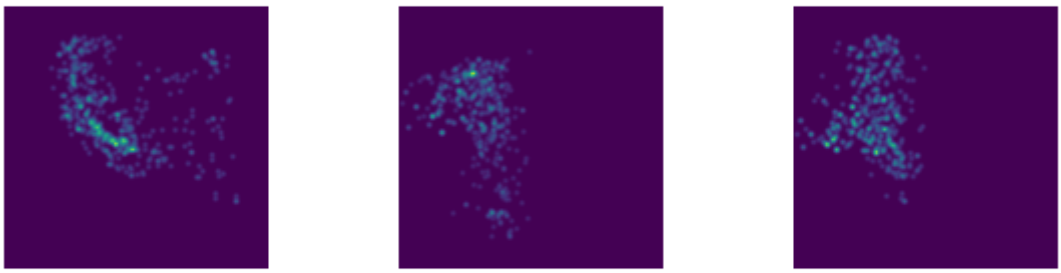


Real GEX (Tissue 2) - X-Y-Z-axis Trhde

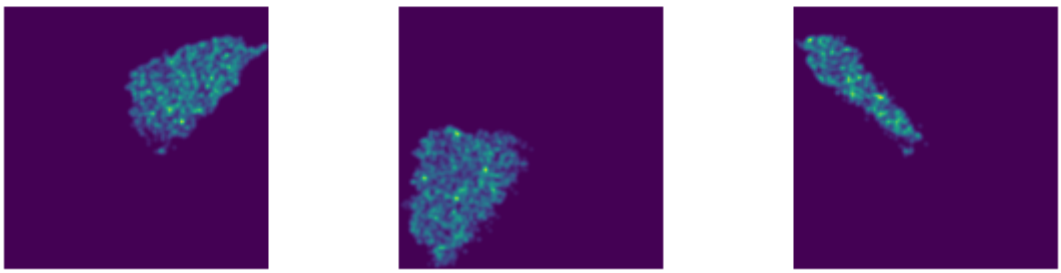


Real GEX (Tissue 3) - X-Y-Z-axis Trhde

Real GEX (Tissue 4) - X-Y-Z-axis Trhde

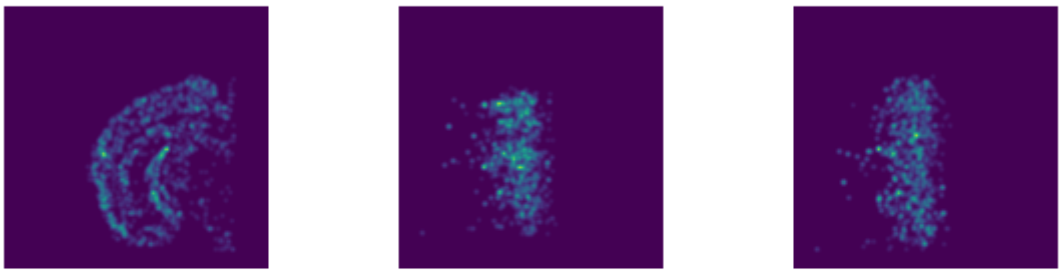


In-Silico GEX - X-Y-Z-axis Trhde

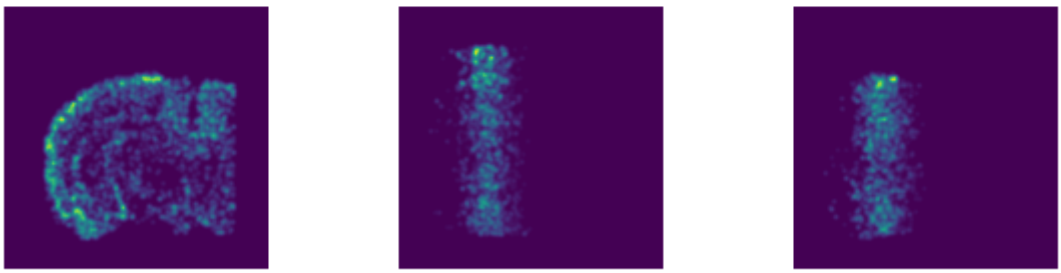


Real GEX Pam

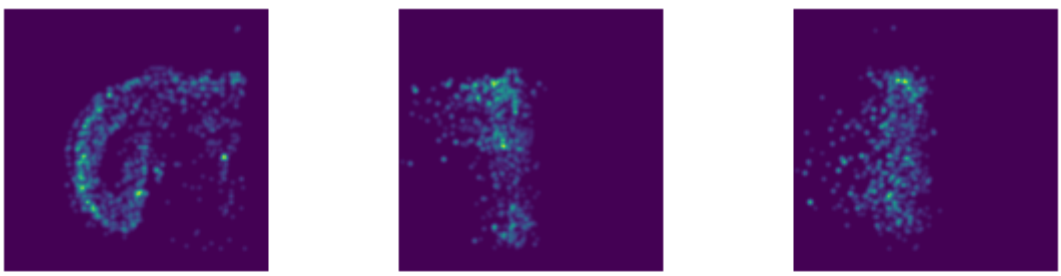
Real GEX (Tissue 1) - X-Y-Z-axis Pam



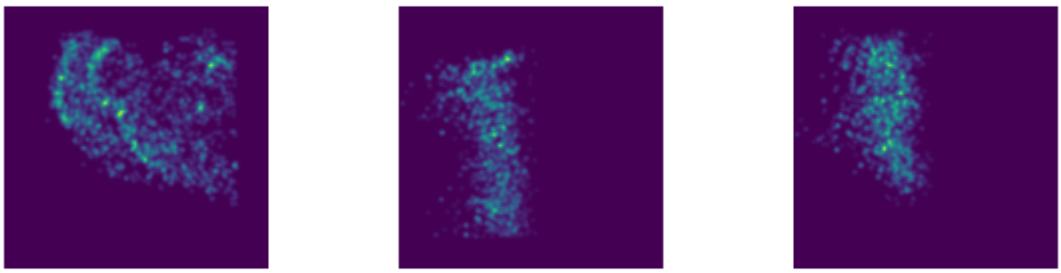
Real GEX (Tissue 2) - X-Y-Z-axis Pam



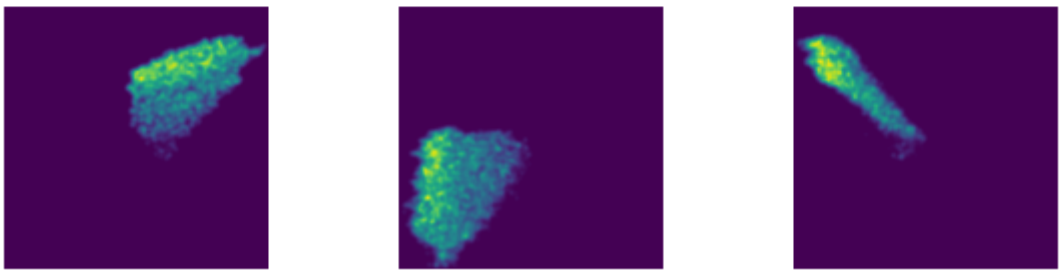
Real GEX (Tissue 3) - X-Y-Z-axis Pam



Real GEX (Tissue 4) - X-Y-Z-axis Pam

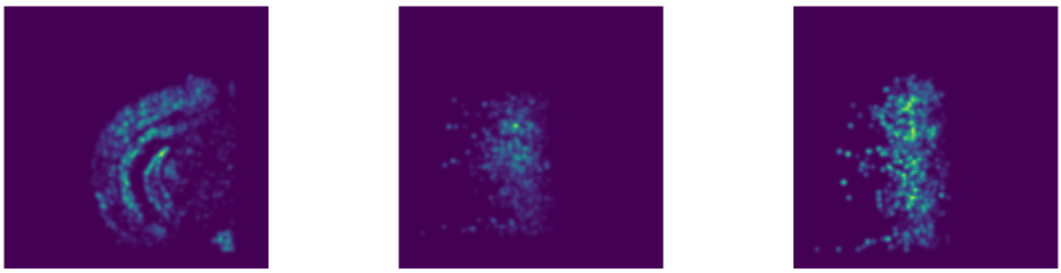


In-Silico GEX - X-Y-Z-axis Pam

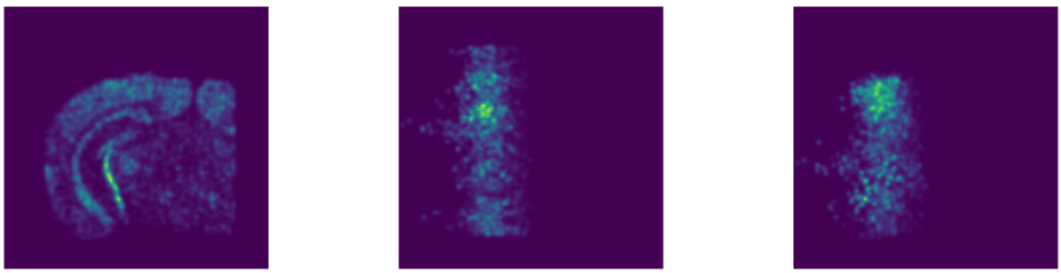


Real GEX Slc24a2

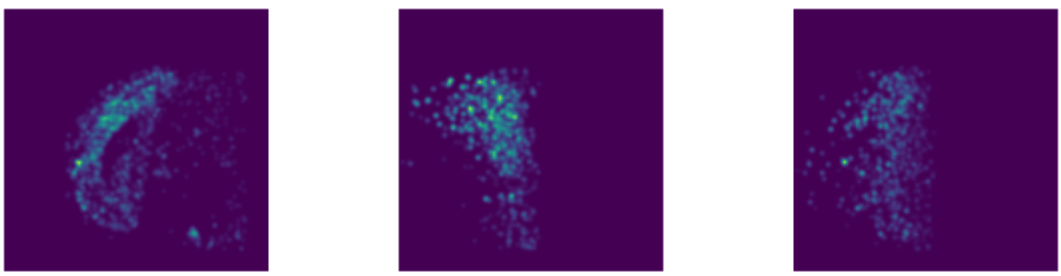
Real GEX (Tissue 1) - X-Y-Z-axis Slc24a2



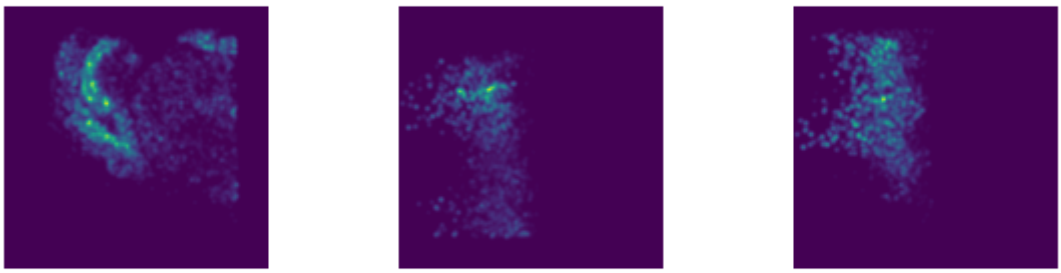
Real GEX (Tissue 2) - X-Y-Z-axis Slc24a2



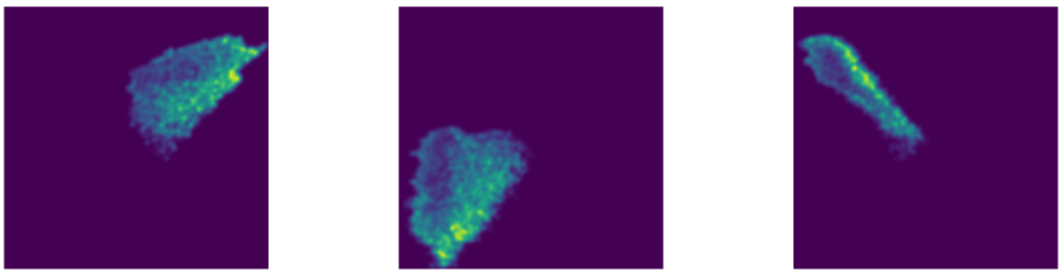
Real GEX (Tissue 3) - X-Y-Z-axis Slc24a2



Real GEX (Tissue 4) - X-Y-Z-axis Slc24a2

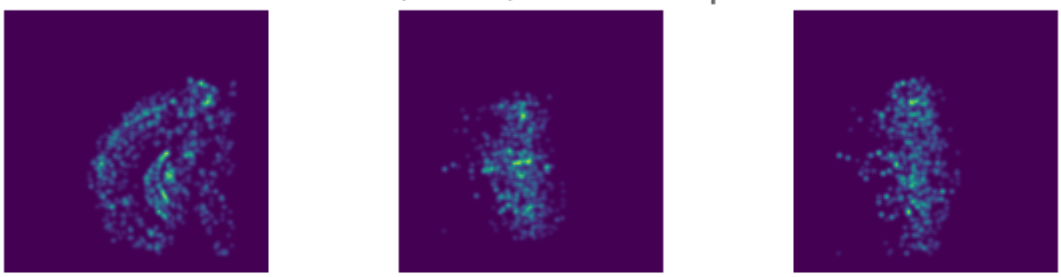


In-Silico GEX - X-Y-Z-axis Slc24a2

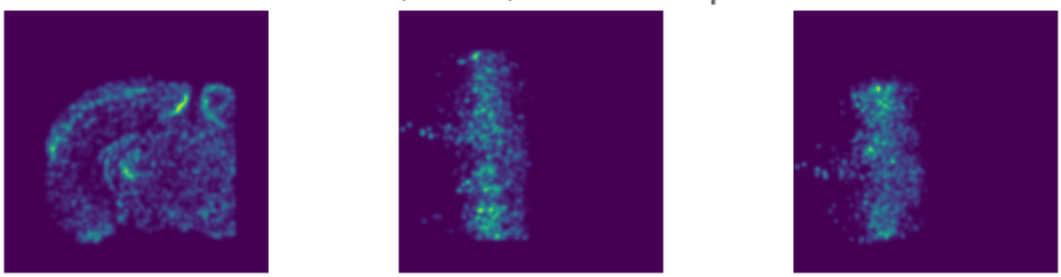


Real GEX Spock3

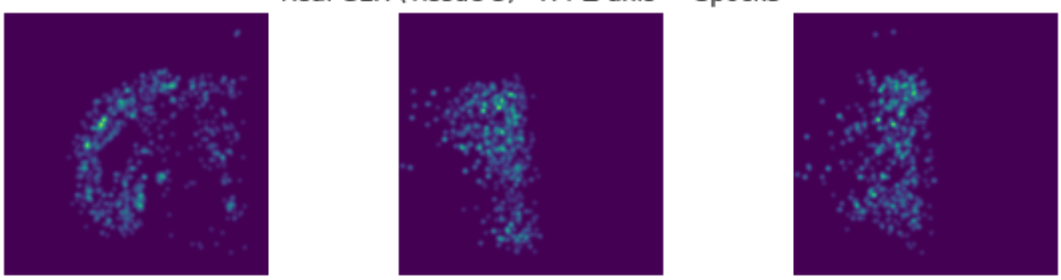
Real GEX (Tissue 1) - X-Y-Z-axis Spock3



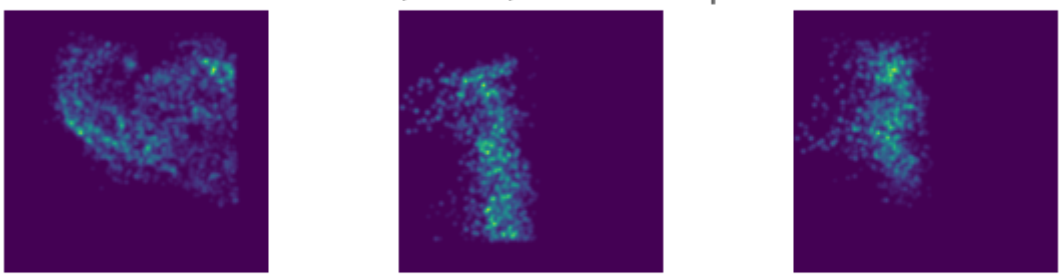
Real GEX (Tissue 2) - X-Y-Z-axis Spock3



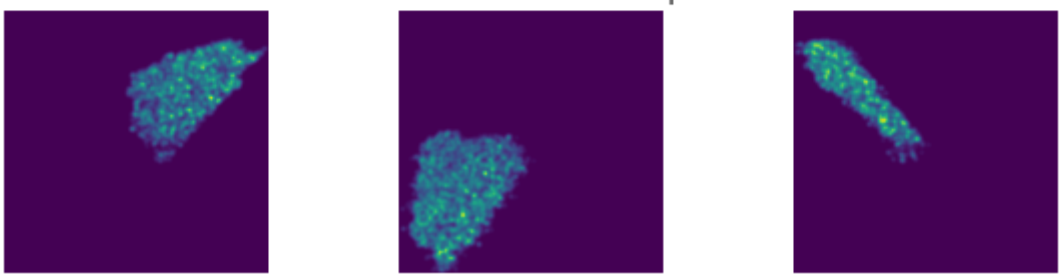
Real GEX (Tissue 3) - X-Y-Z-axis Spock3



Real GEX (Tissue 4) - X-Y-Z-axis Spock3

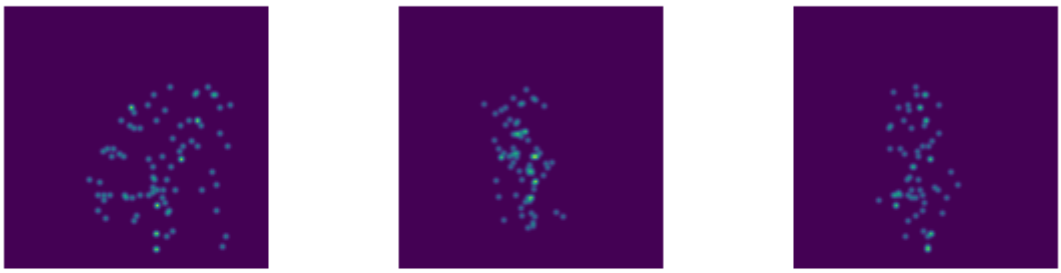


In-Silico GEX - X-Y-Z-axis Spock3

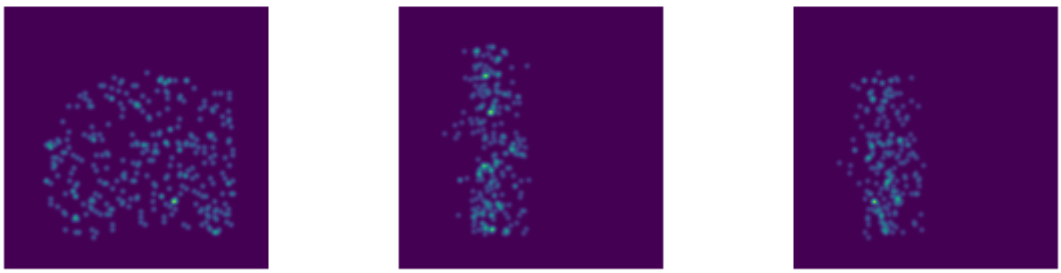


Real GEX Tmtc2

Real GEX (Tissue 1) - X-Y-Z-axis Tmtc2

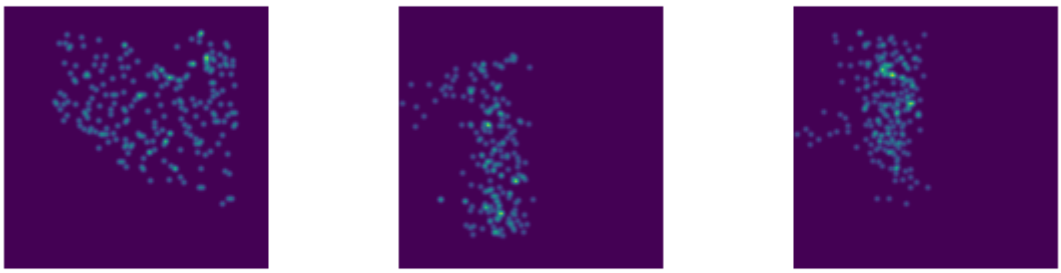


Real GEX (Tissue 2) - X-Y-Z-axis Tmtc2

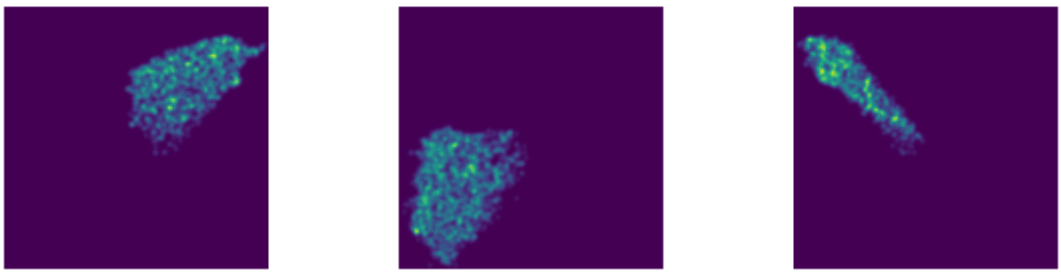


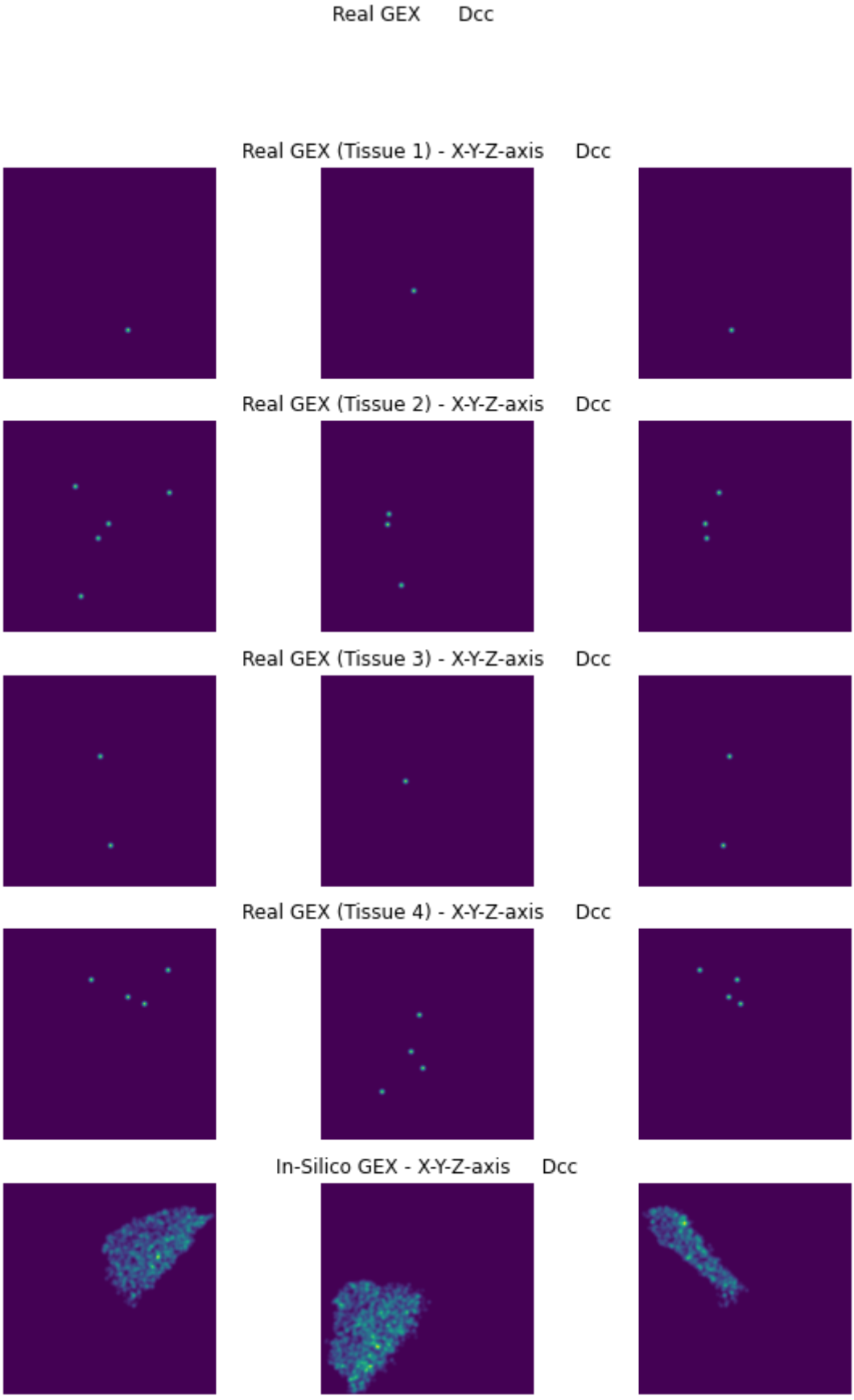
Real GEX (Tissue 3) - X-Y-Z-axis Tmtc2

Real GEX (Tissue 4) - X-Y-Z-axis Tmtc2



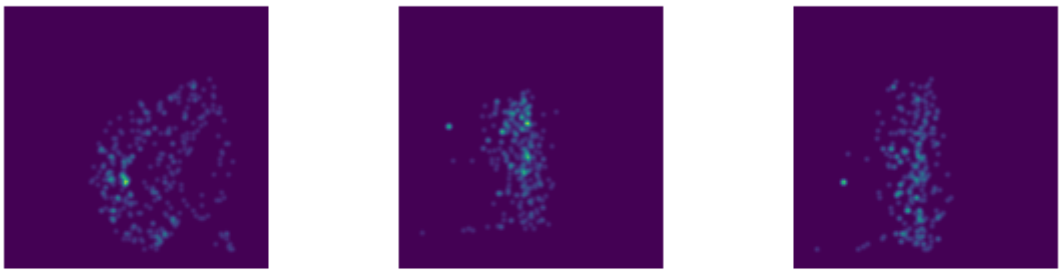
In-Silico GEX - X-Y-Z-axis Tmtc2



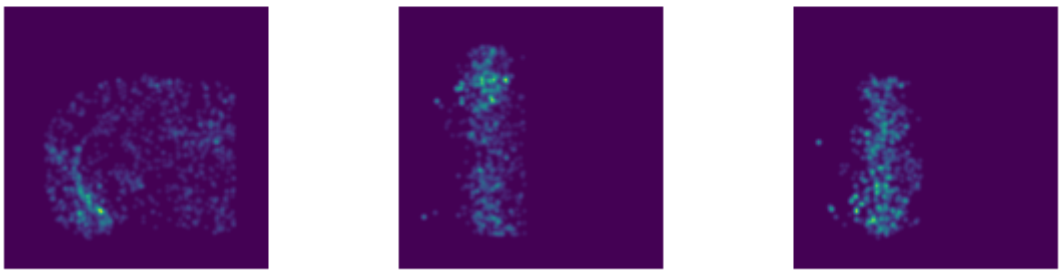

Real GEX Dcc
Real GEX (Tissue 1) - X-Y-Z-axis Dcc
Real GEX (Tissue 2) - X-Y-Z-axis Dcc
Real GEX (Tissue 3) - X-Y-Z-axis Dcc
Real GEX (Tissue 4) - X-Y-Z-axis Dcc
In-Silico GEX - X-Y-Z-axis Dcc

Real GEX Grin3a

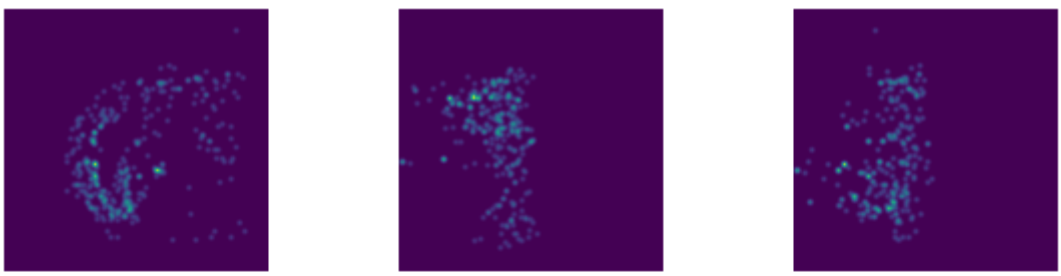
Real GEX (Tissue 1) - X-Y-Z-axis Grin3a



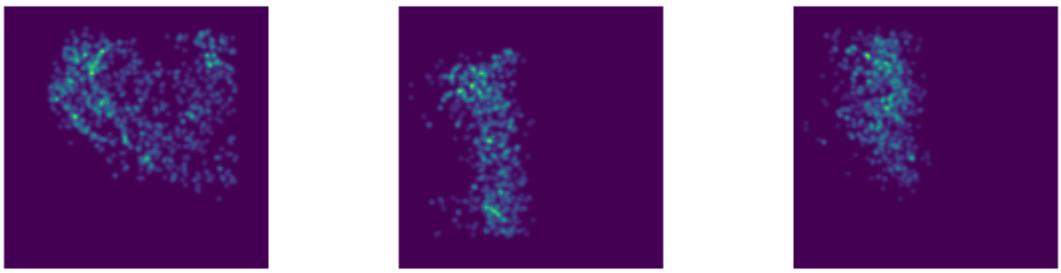
Real GEX (Tissue 2) - X-Y-Z-axis Grin3a



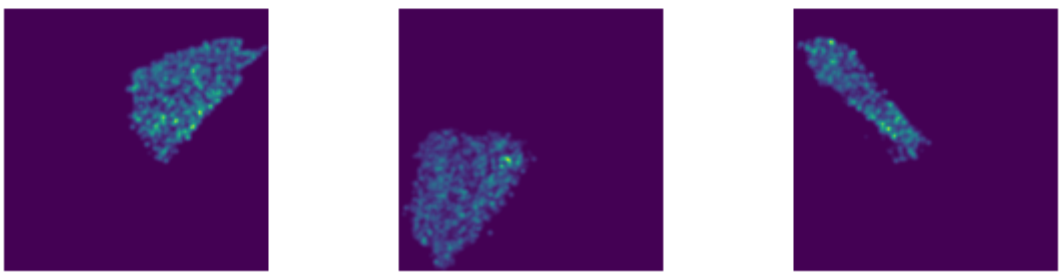
Real GEX (Tissue 3) - X-Y-Z-axis Grin3a



Real GEX (Tissue 4) - X-Y-Z-axis Grin3a

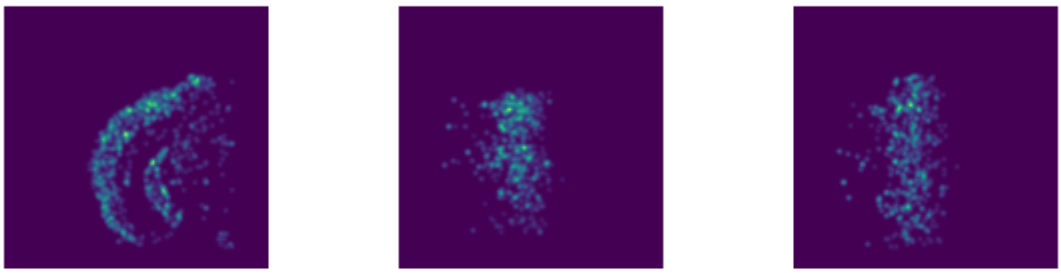


In-Silico GEX - X-Y-Z-axis Grin3a

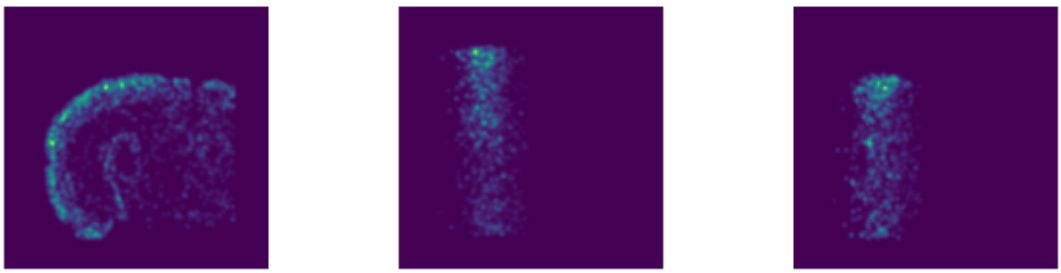


Real GEX Car10

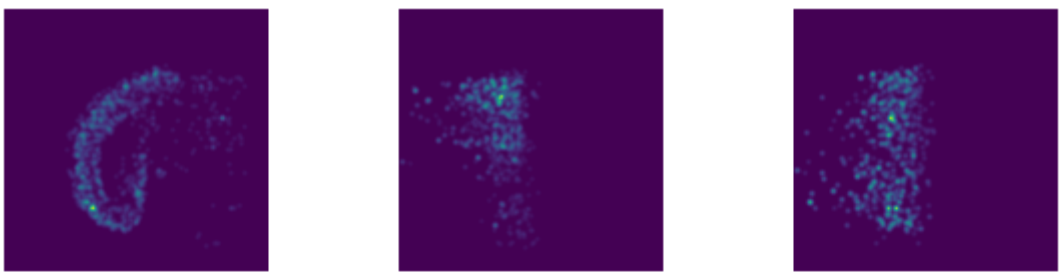
Real GEX (Tissue 1) - X-Y-Z-axis Car10



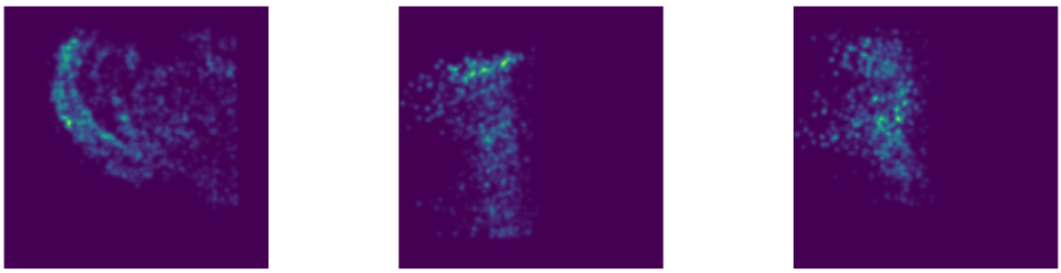
Real GEX (Tissue 2) - X-Y-Z-axis Car10



Real GEX (Tissue 3) - X-Y-Z-axis Car10



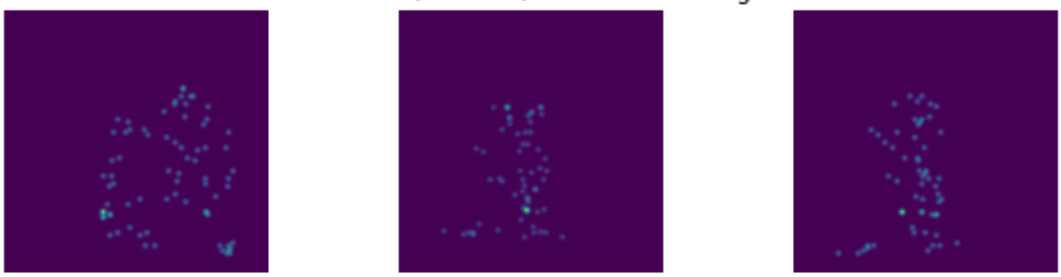
Real GEX (Tissue 4) - X-Y-Z-axis Car10



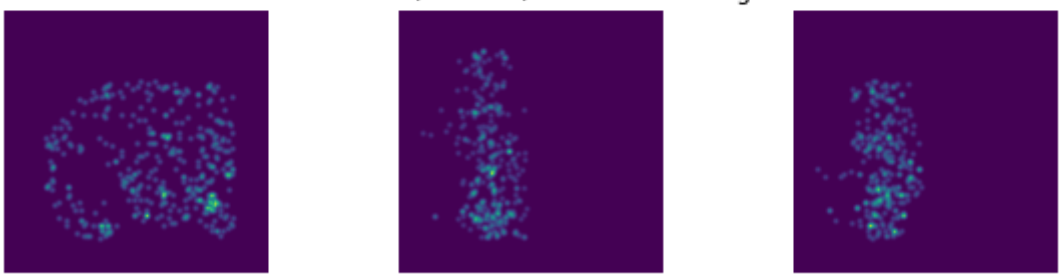
In-Silico GEX - X-Y-Z-axis Car10

Real GEX Nrg1

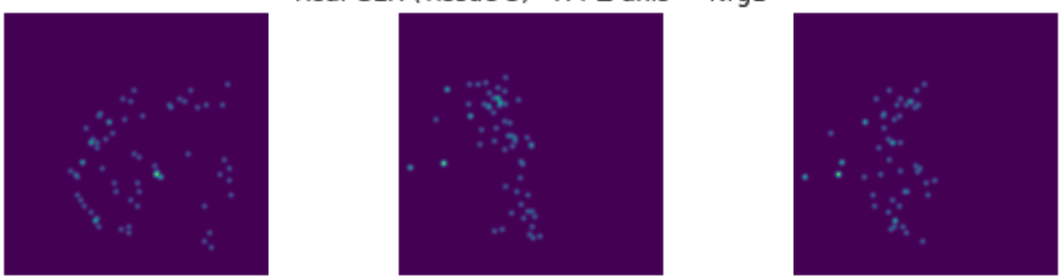
Real GEX (Tissue 1) - X-Y-Z-axis Nrg1



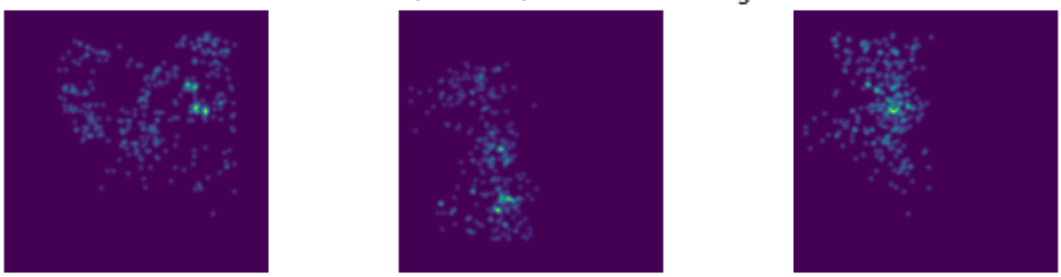
Real GEX (Tissue 2) - X-Y-Z-axis Nrg1



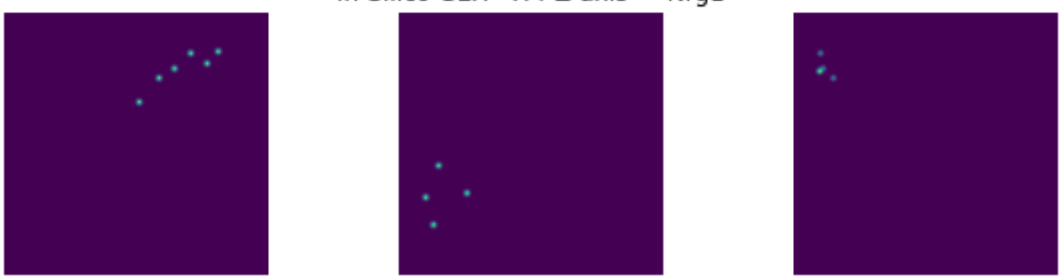
Real GEX (Tissue 3) - X-Y-Z-axis Nrg1



Real GEX (Tissue 4) - X-Y-Z-axis Nrg1

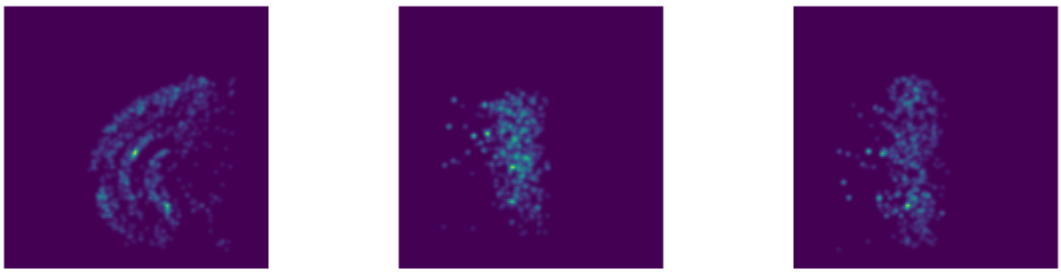


In-Silico GEX - X-Y-Z-axis Nrg1

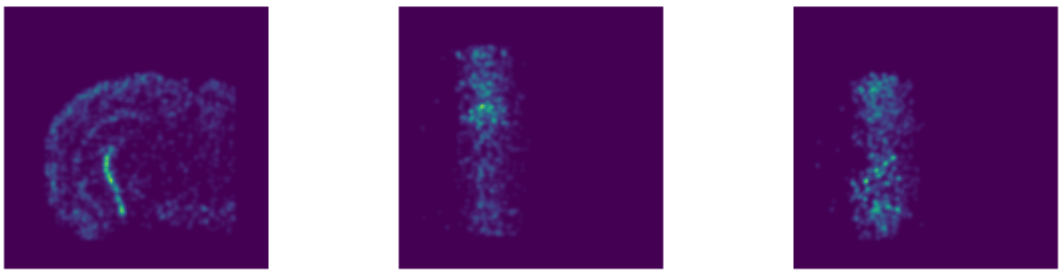


Real GEX Prkca

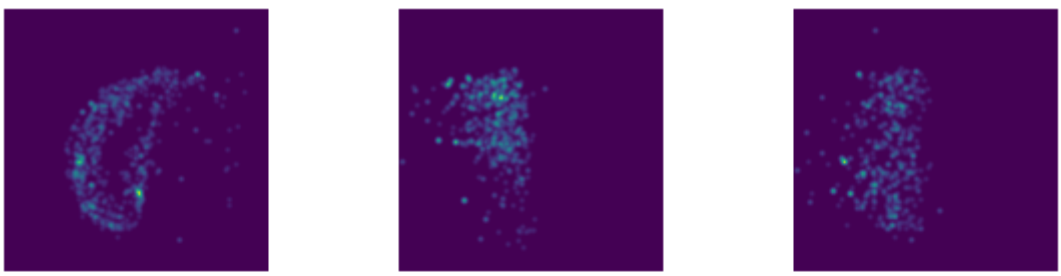
Real GEX (Tissue 1) - X-Y-Z-axis Prkca



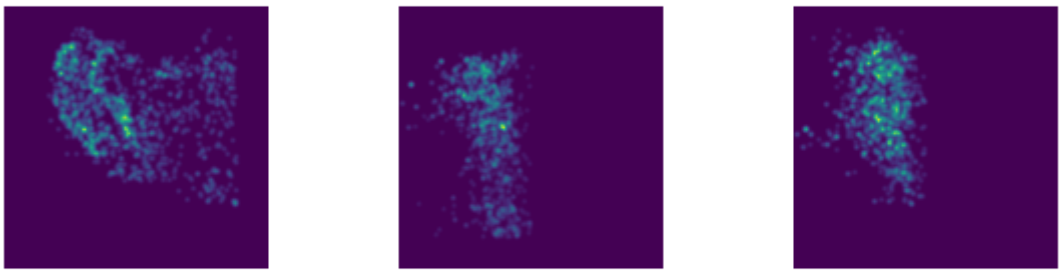
Real GEX (Tissue 2) - X-Y-Z-axis Prkca



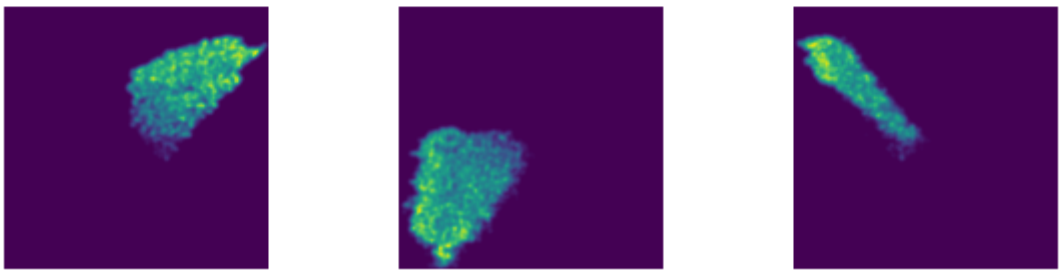
Real GEX (Tissue 3) - X-Y-Z-axis Prkca

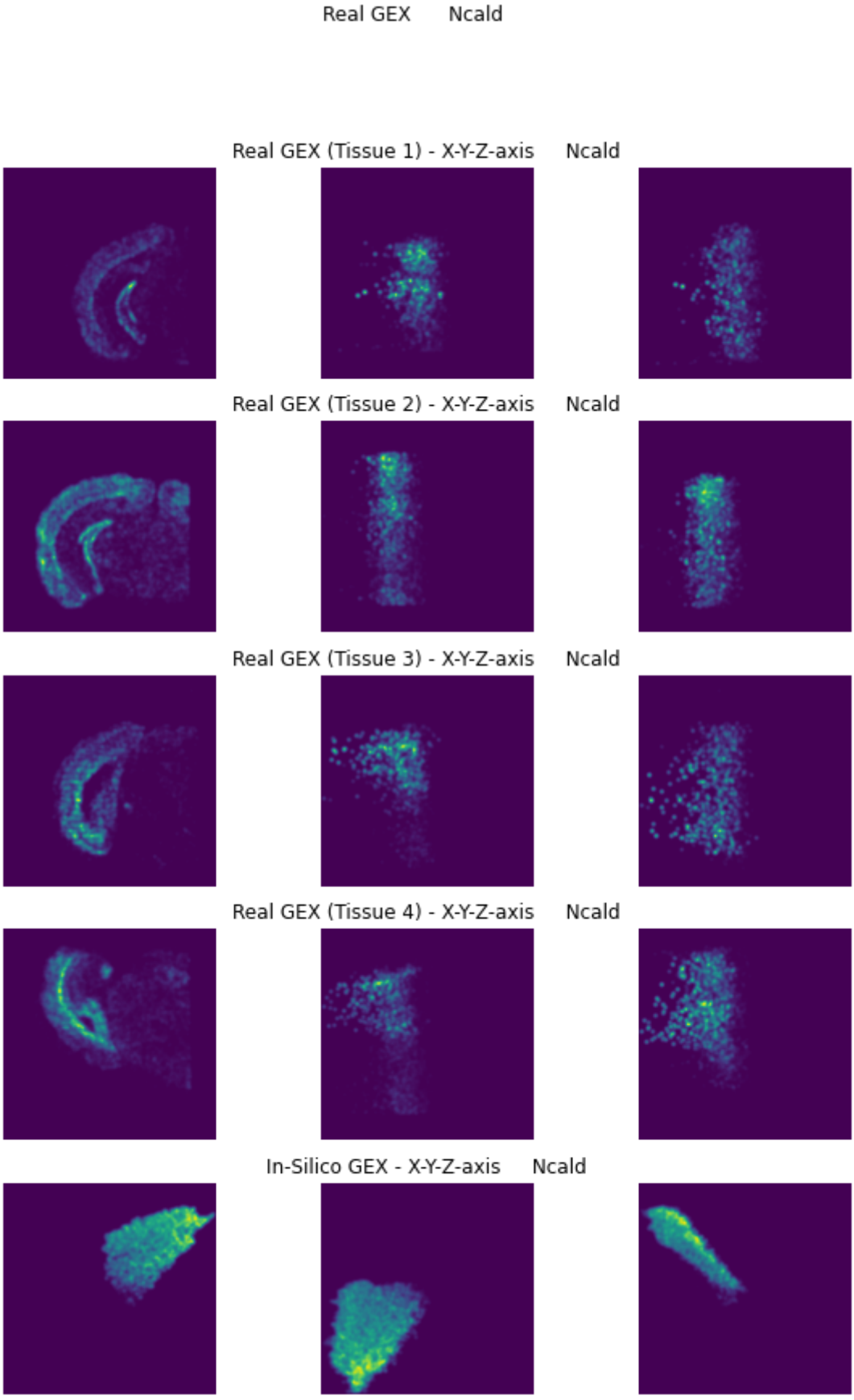


Real GEX (Tissue 4) - X-Y-Z-axis Prkca



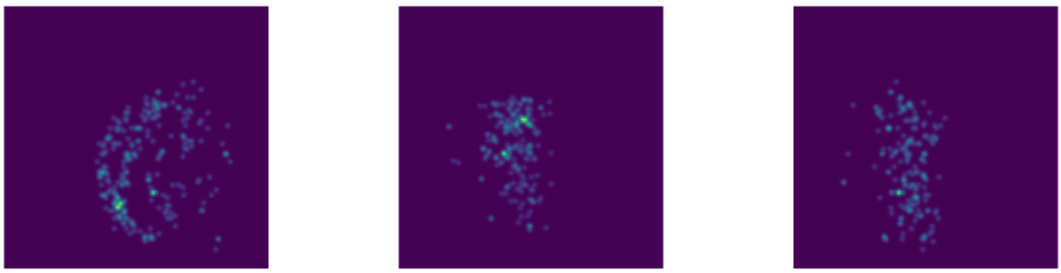
In-Silico GEX - X-Y-Z-axis Prkca



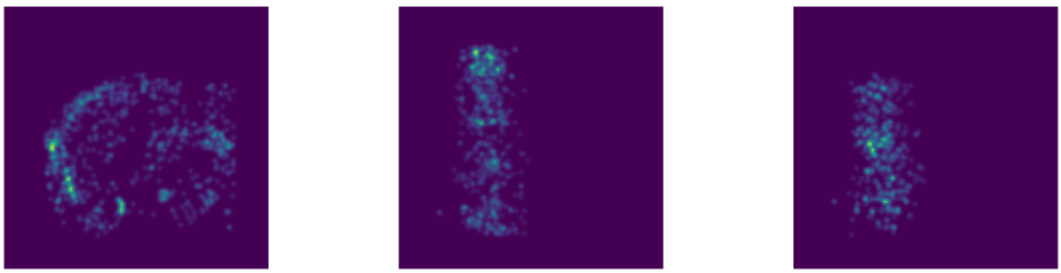

Real GEX
Ncald
Real GEX (Tissue 1) - X-Y-Z-axis
Ncald
Real GEX (Tissue 2) - X-Y-Z-axis
Ncald
Real GEX (Tissue 3) - X-Y-Z-axis
Ncald
Real GEX (Tissue 4) - X-Y-Z-axis
Ncald
In-Silico GEX - X-Y-Z-axis
Ncald

Real GEX Htr2a

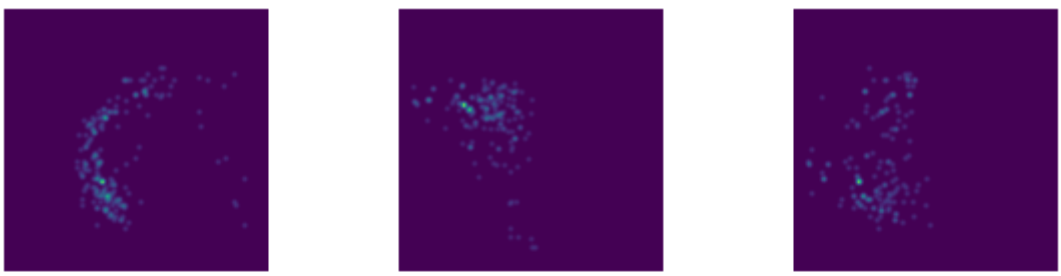
Real GEX (Tissue 1) - X-Y-Z-axis Htr2a



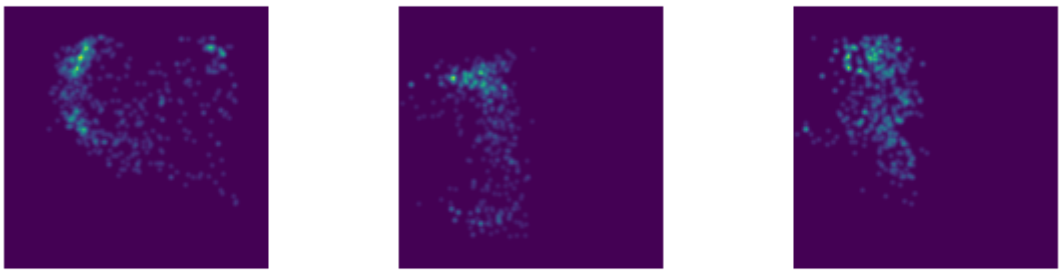
Real GEX (Tissue 2) - X-Y-Z-axis Htr2a



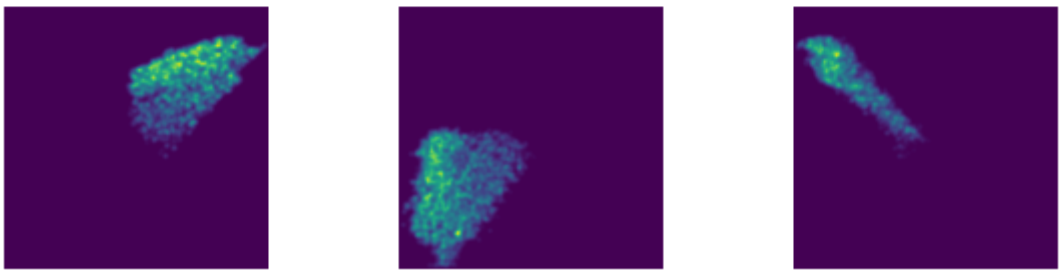
Real GEX (Tissue 3) - X-Y-Z-axis Htr2a



Real GEX (Tissue 4) - X-Y-Z-axis Htr2a

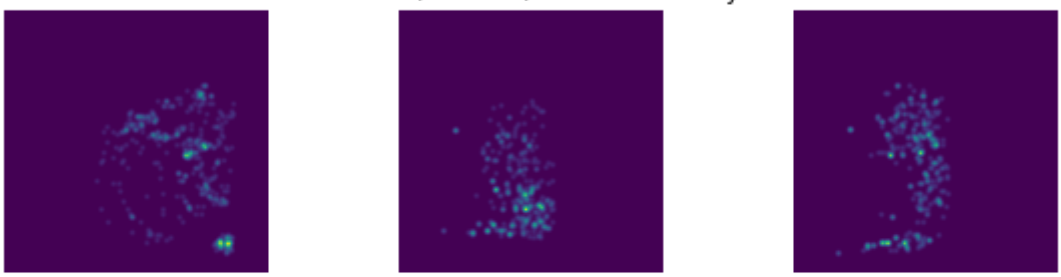


In-Silico GEX - X-Y-Z-axis Htr2a

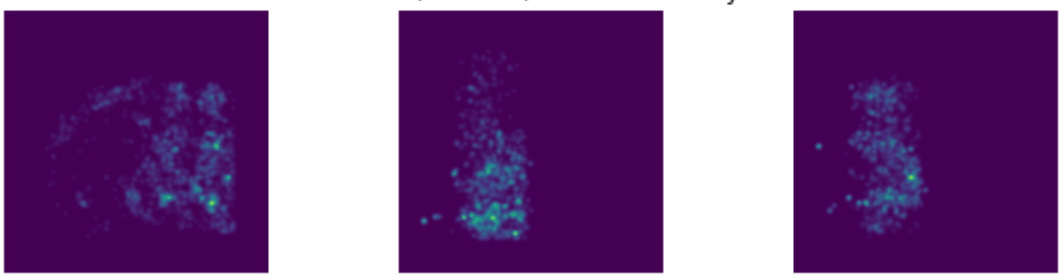


Real GEX Syt2

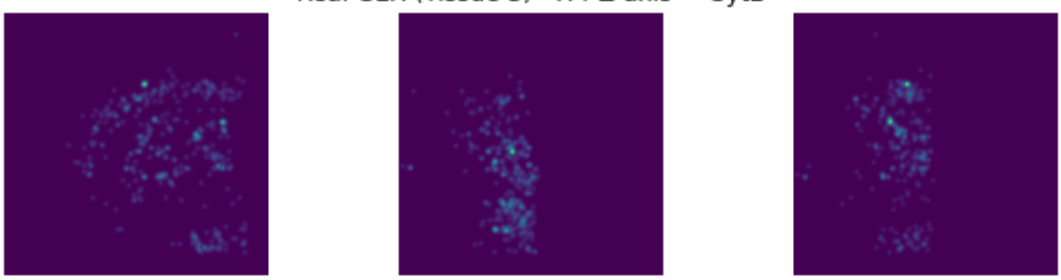
Real GEX (Tissue 1) - X-Y-Z-axis Syt2



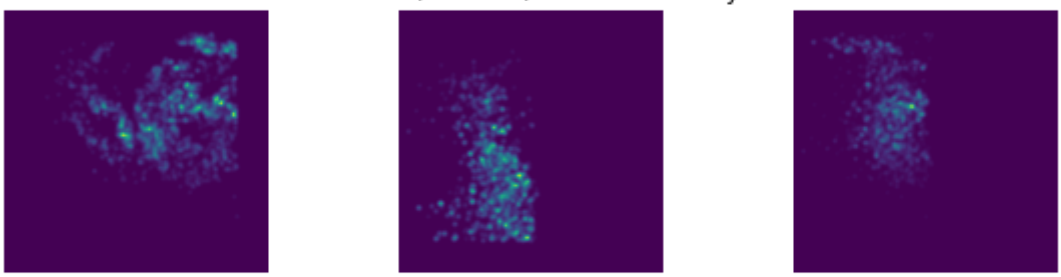
Real GEX (Tissue 2) - X-Y-Z-axis Syt2



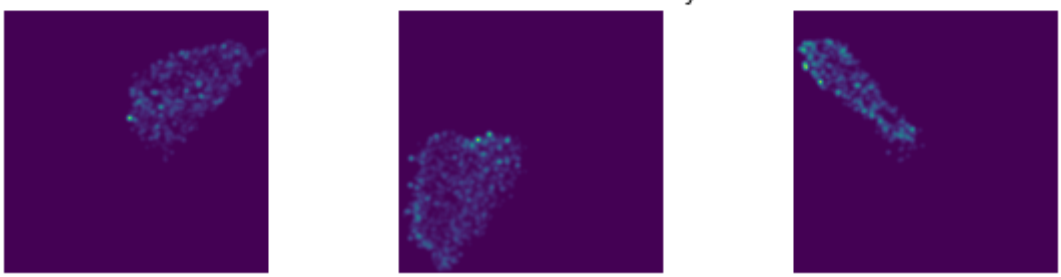
Real GEX (Tissue 3) - X-Y-Z-axis Syt2



Real GEX (Tissue 4) - X-Y-Z-axis Syt2

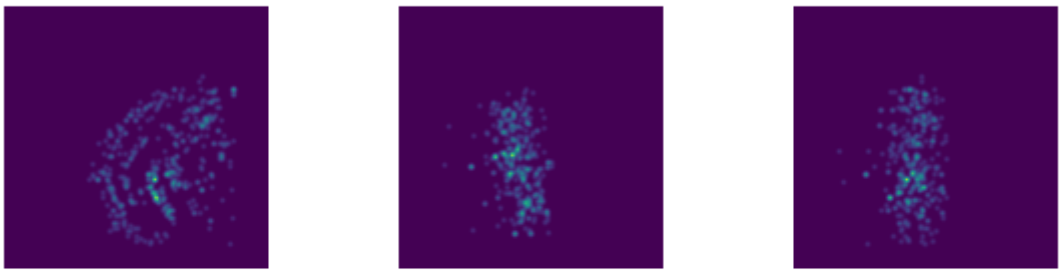


In-Silico GEX - X-Y-Z-axis Syt2

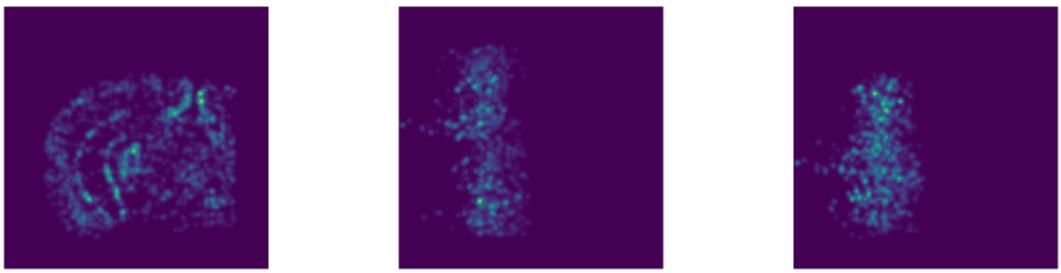


Real GEX Nell1

Real GEX (Tissue 1) - X-Y-Z-axis Nell1

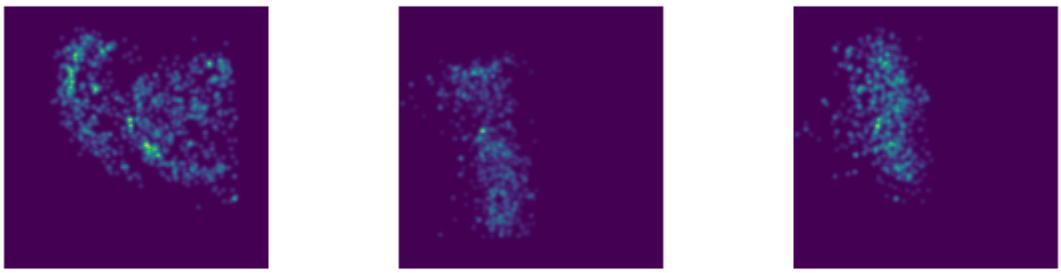


Real GEX (Tissue 2) - X-Y-Z-axis Nell1

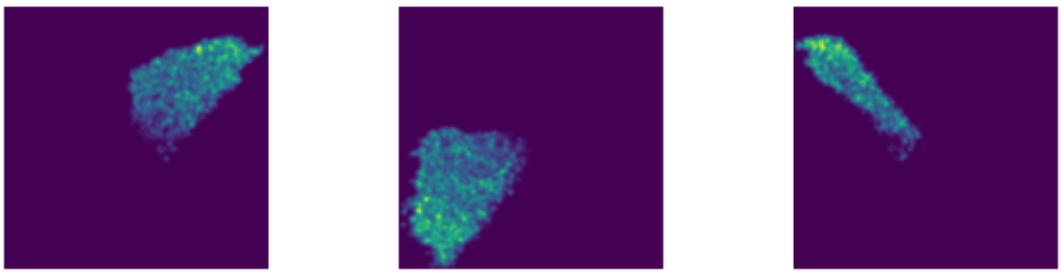


Real GEX (Tissue 3) - X-Y-Z-axis Nell1

Real GEX (Tissue 4) - X-Y-Z-axis Nell1

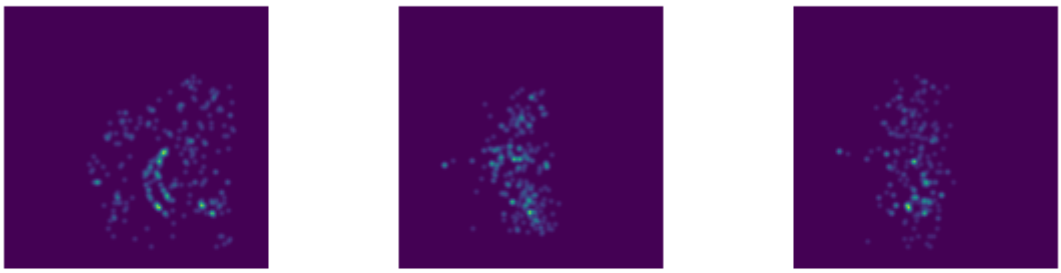


In-Silico GEX - X-Y-Z-axis Nell1

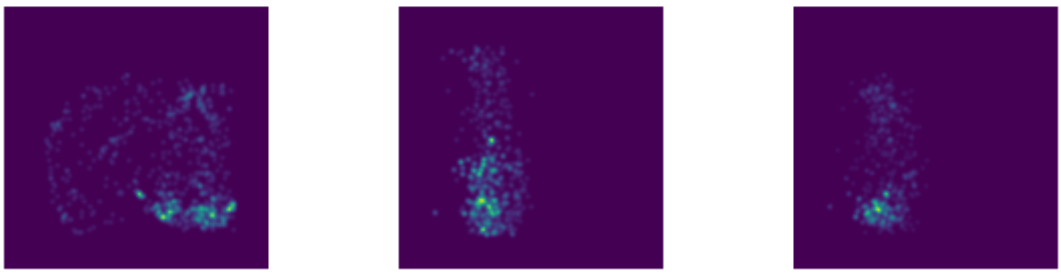


Real GEX Sv2c

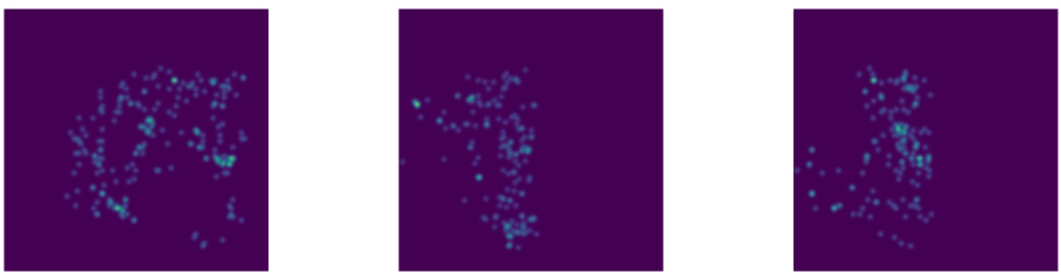
Real GEX (Tissue 1) - X-Y-Z-axis Sv2c



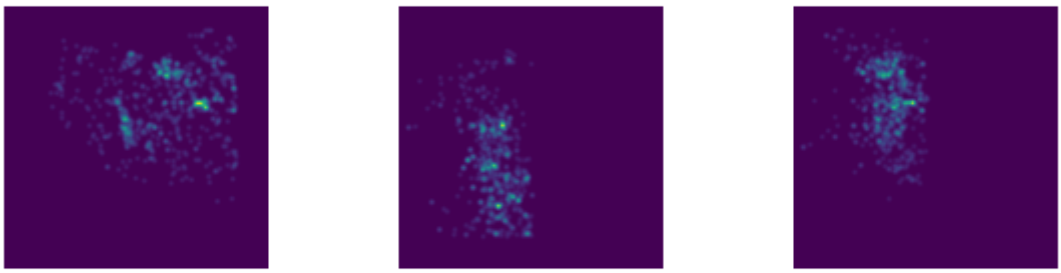
Real GEX (Tissue 2) - X-Y-Z-axis Sv2c



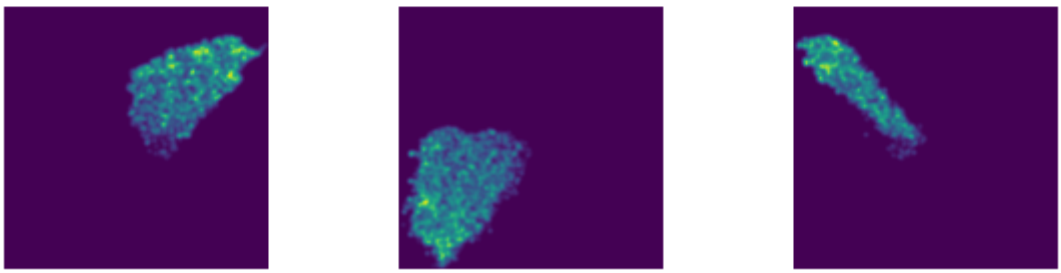
Real GEX (Tissue 3) - X-Y-Z-axis Sv2c



Real GEX (Tissue 4) - X-Y-Z-axis Sv2c

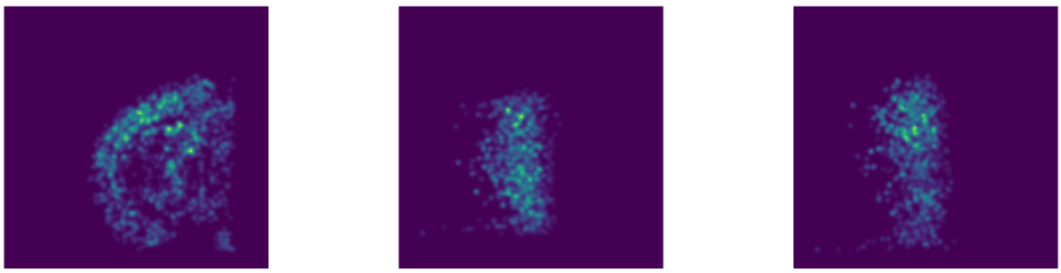


In-Silico GEX - X-Y-Z-axis Sv2c

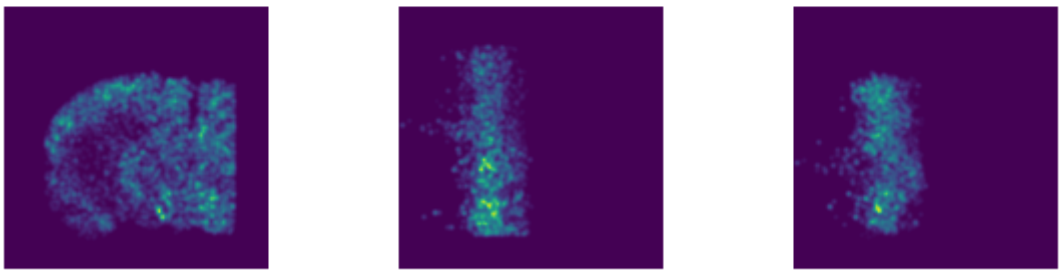


Real GEX Rcan2

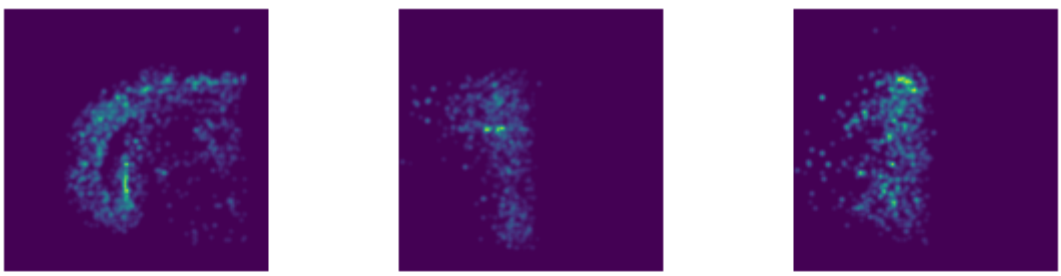
Real GEX (Tissue 1) - X-Y-Z-axis Rcan2



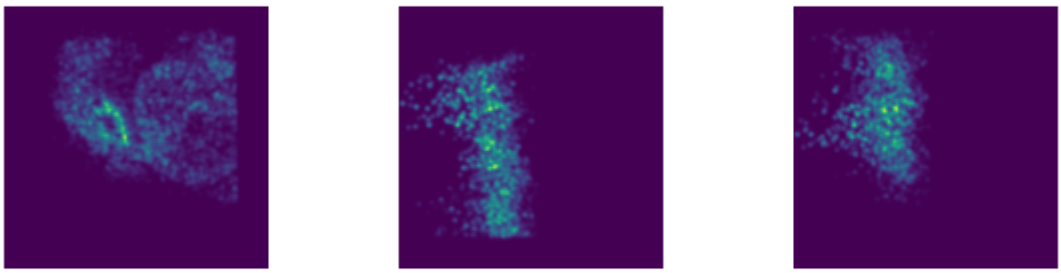
Real GEX (Tissue 2) - X-Y-Z-axis Rcan2



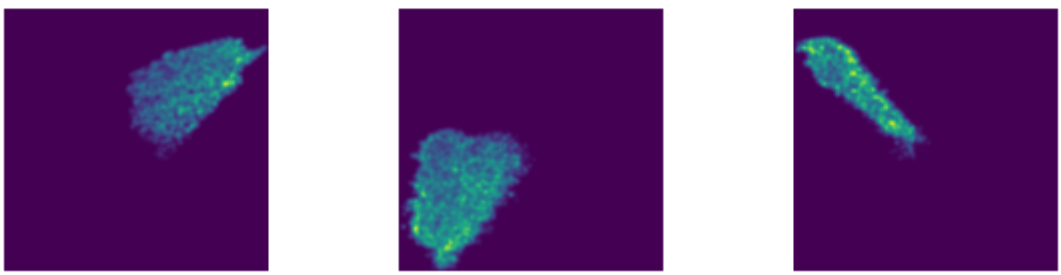
Real GEX (Tissue 3) - X-Y-Z-axis Rcan2

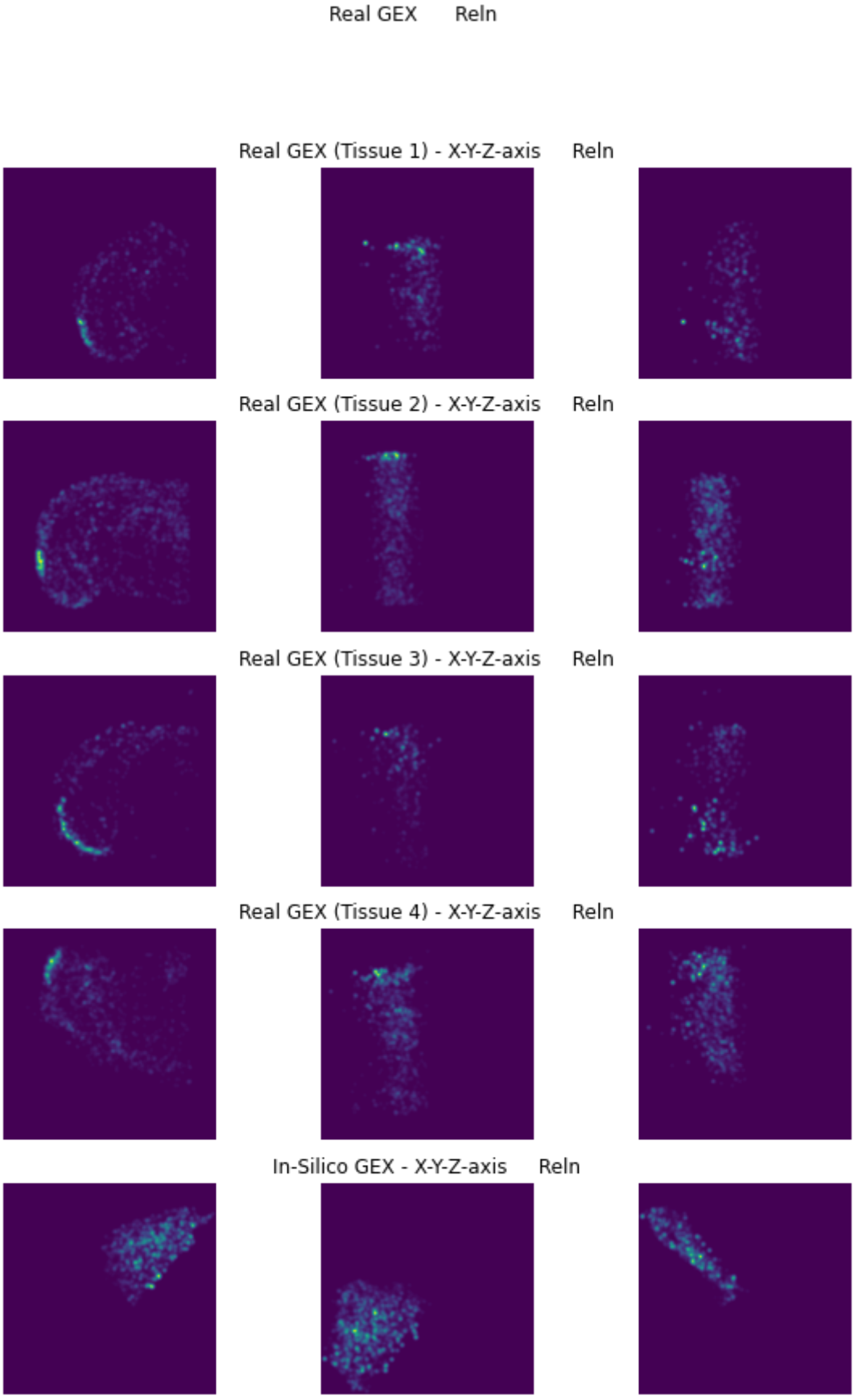


Real GEX (Tissue 4) - X-Y-Z-axis Rcan2



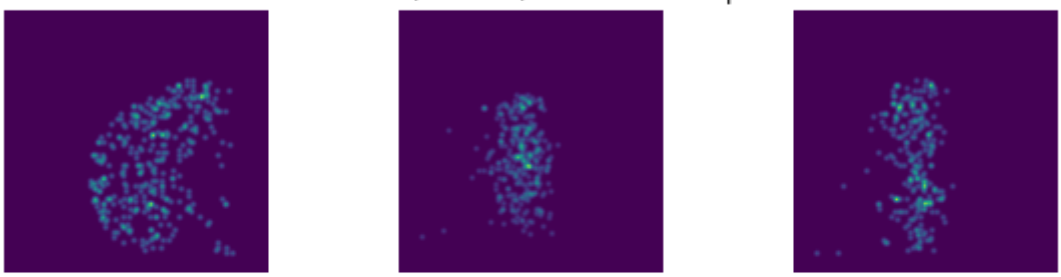
In-Silico GEX - X-Y-Z-axis Rcan2



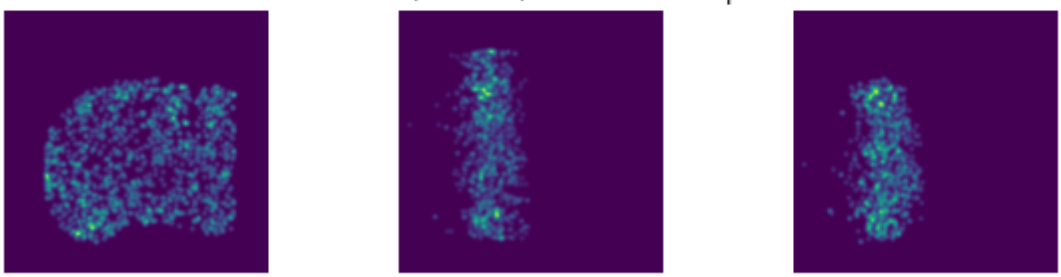

Real GEX
Reln
Real GEX (Tissue 1) - X-Y-Z-axis
Reln
Real GEX (Tissue 2) - X-Y-Z-axis
Reln
Real GEX (Tissue 3) - X-Y-Z-axis
Reln
Real GEX (Tissue 4) - X-Y-Z-axis
Reln
In-Silico GEX - X-Y-Z-axis
Reln

Real GEX Gpc5

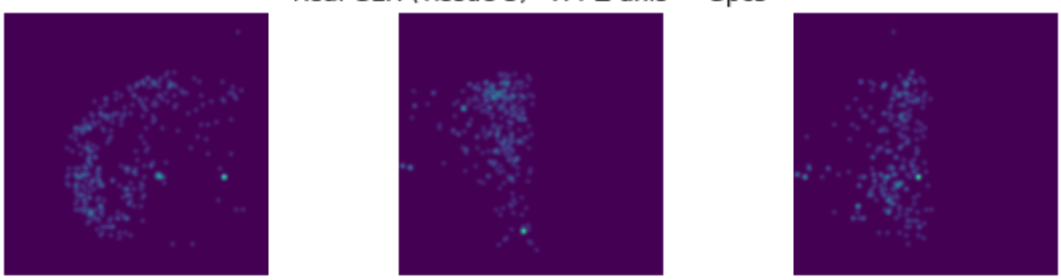
Real GEX (Tissue 1) - X-Y-Z-axis Gpc5



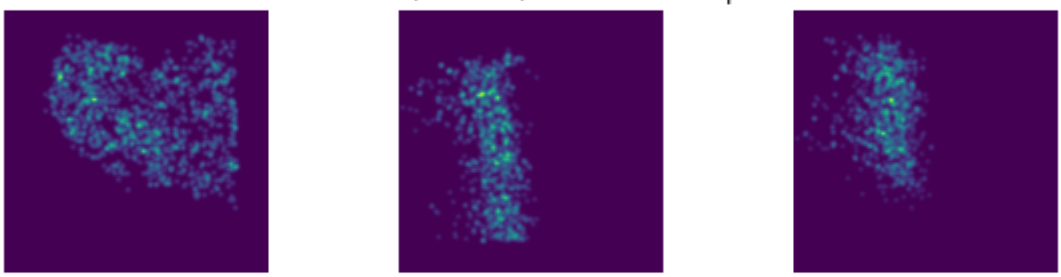
Real GEX (Tissue 2) - X-Y-Z-axis Gpc5



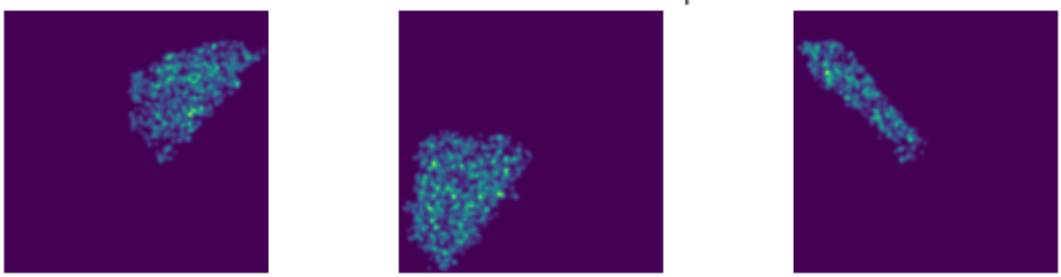
Real GEX (Tissue 3) - X-Y-Z-axis Gpc5



Real GEX (Tissue 4) - X-Y-Z-axis Gpc5

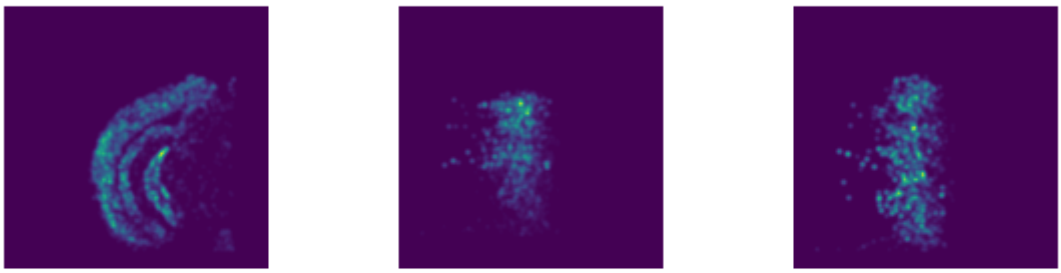


In-Silico GEX - X-Y-Z-axis Gpc5

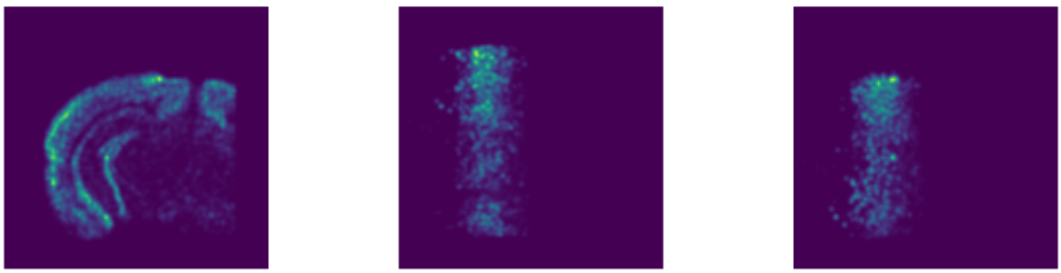


Real GEX Lmo4

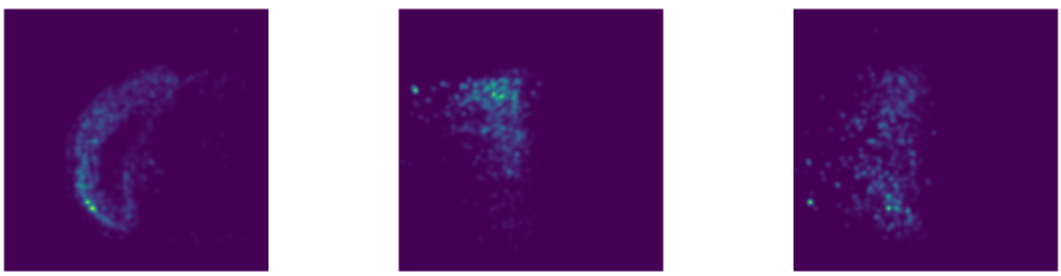
Real GEX (Tissue 1) - X-Y-Z-axis Lmo4



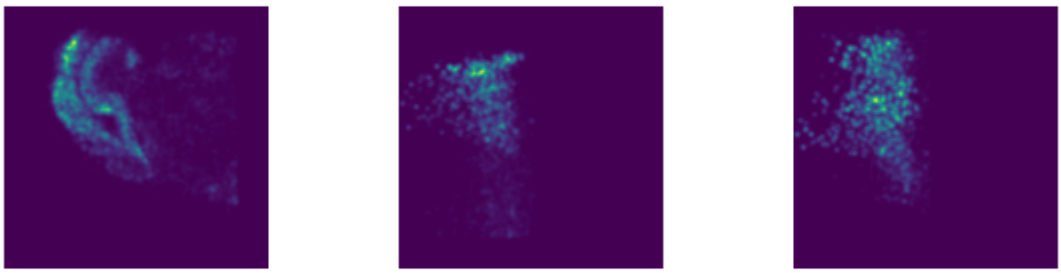
Real GEX (Tissue 2) - X-Y-Z-axis Lmo4



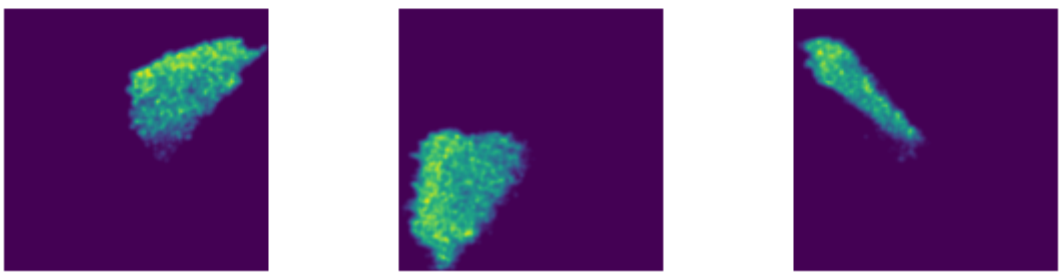
Real GEX (Tissue 3) - X-Y-Z-axis Lmo4



Real GEX (Tissue 4) - X-Y-Z-axis Lmo4

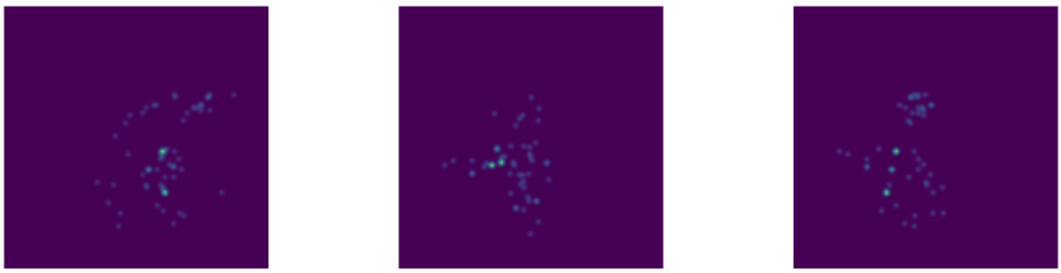


In-Silico GEX - X-Y-Z-axis Lmo4

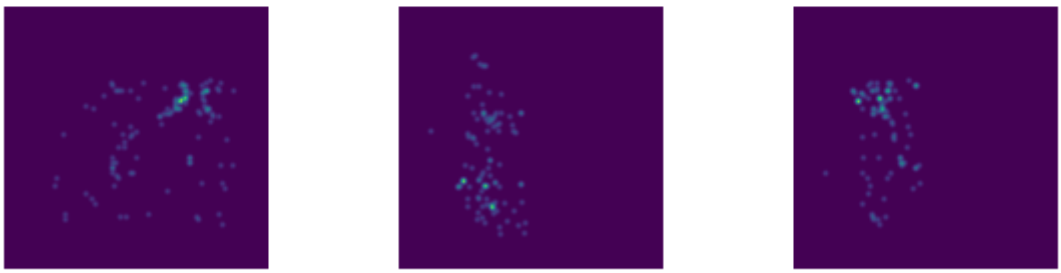


Real GEX Scnn1a

Real GEX (Tissue 1) - X-Y-Z-axis Scnn1a

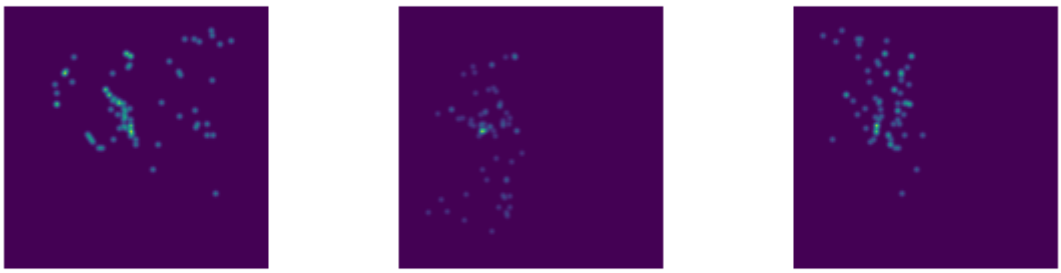


Real GEX (Tissue 2) - X-Y-Z-axis Scnn1a

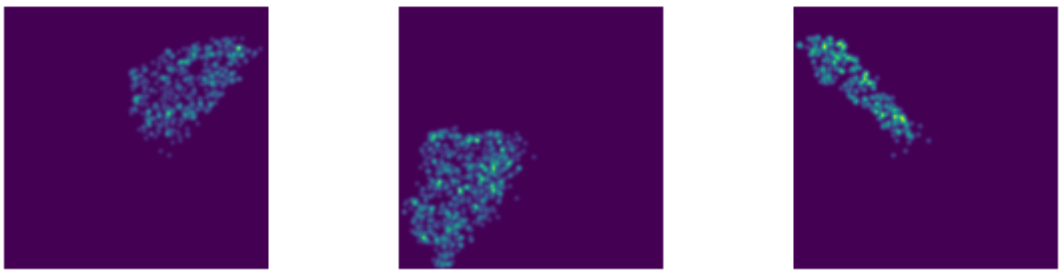


Real GEX (Tissue 3) - X-Y-Z-axis Scnn1a

Real GEX (Tissue 4) - X-Y-Z-axis Scnn1a

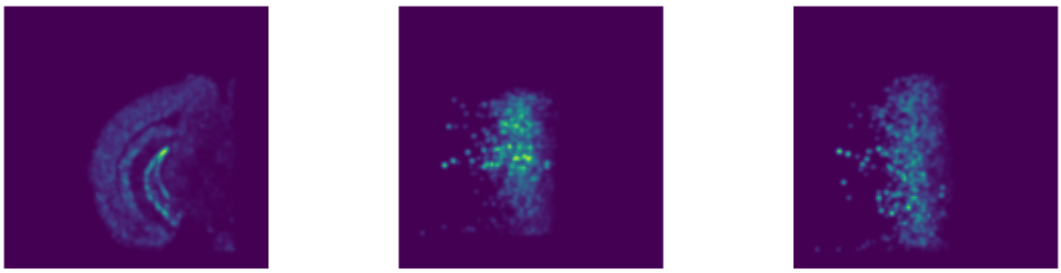


In-Silico GEX - X-Y-Z-axis Scnn1a

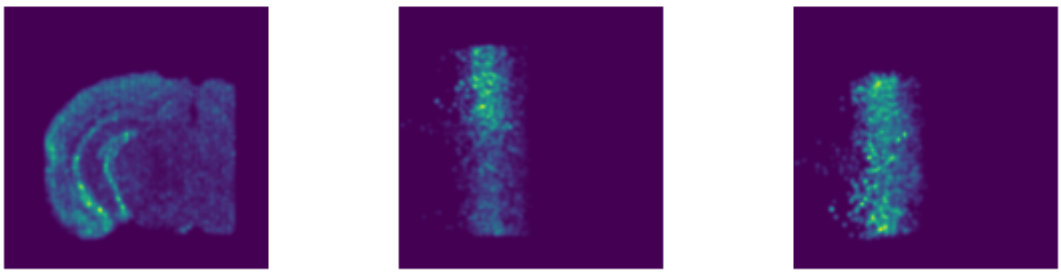


Real GEX Actb

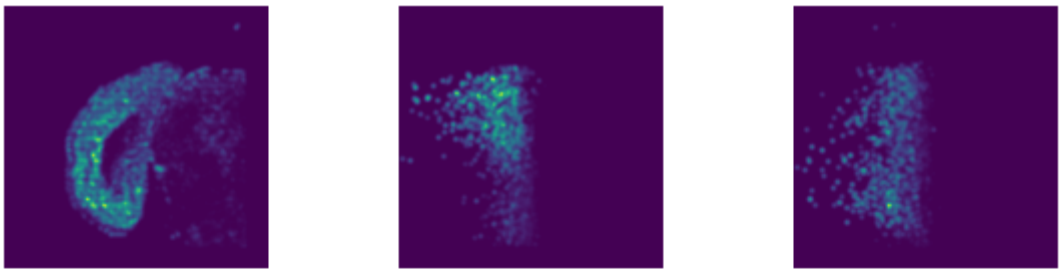
Real GEX (Tissue 1) - X-Y-Z-axis Actb



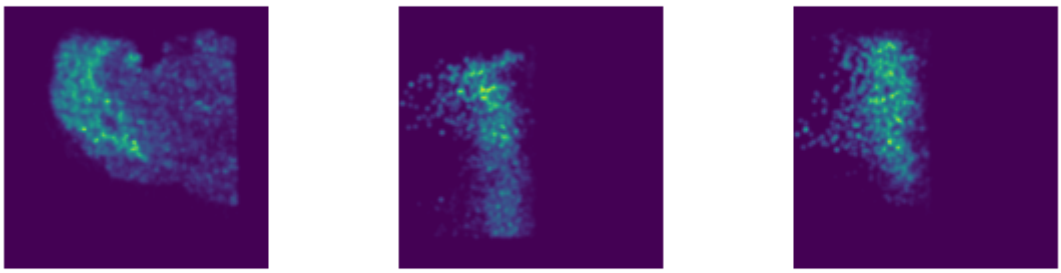
Real GEX (Tissue 2) - X-Y-Z-axis Actb



Real GEX (Tissue 3) - X-Y-Z-axis Actb



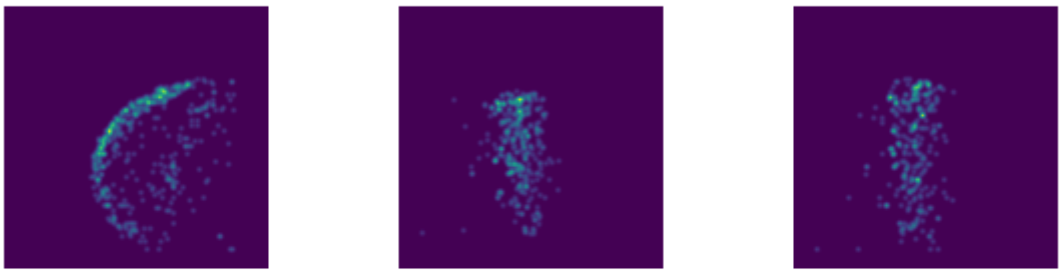
Real GEX (Tissue 4) - X-Y-Z-axis Actb



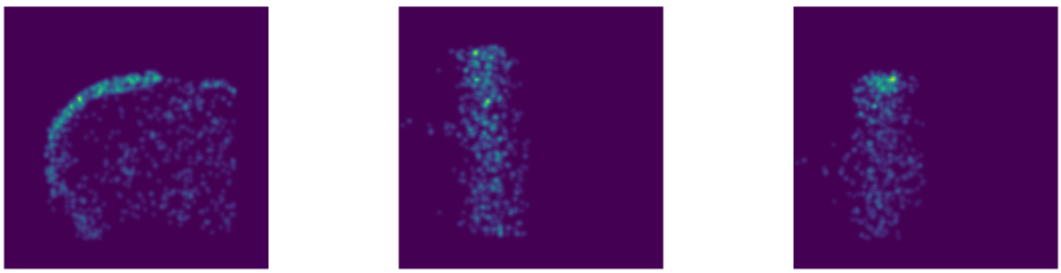
In-Silico GEX - X-Y-Z-axis Actb

Real GEX Cux2

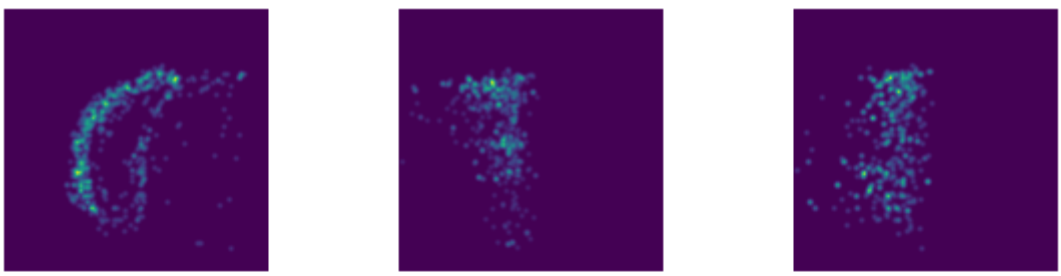
Real GEX (Tissue 1) - X-Y-Z-axis Cux2



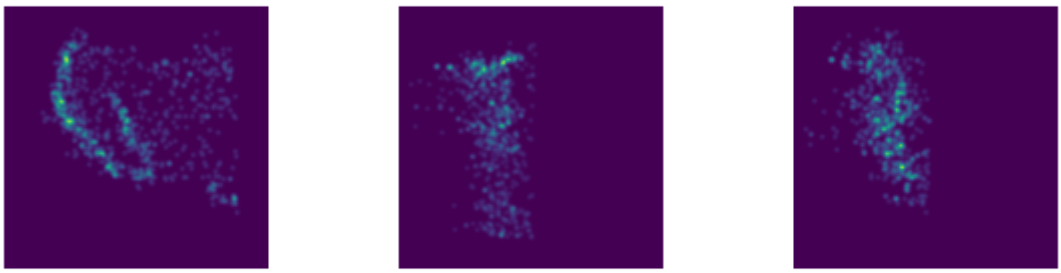
Real GEX (Tissue 2) - X-Y-Z-axis Cux2



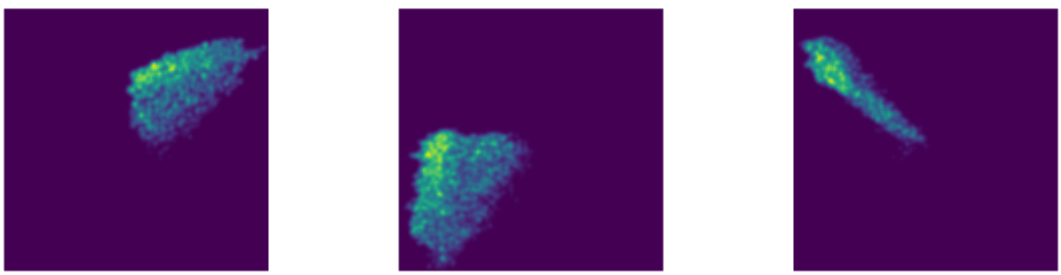
Real GEX (Tissue 3) - X-Y-Z-axis Cux2

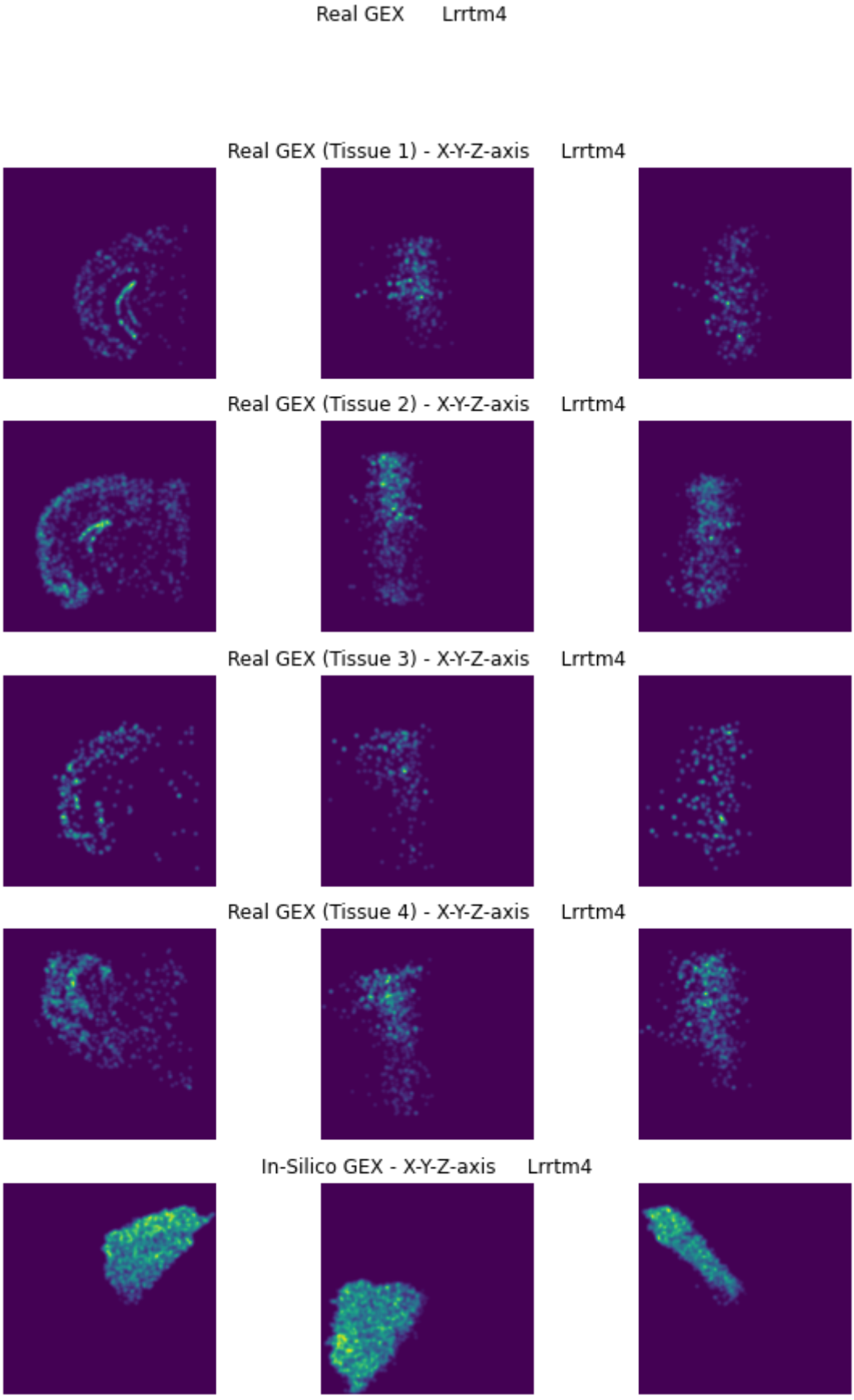


Real GEX (Tissue 4) - X-Y-Z-axis Cux2



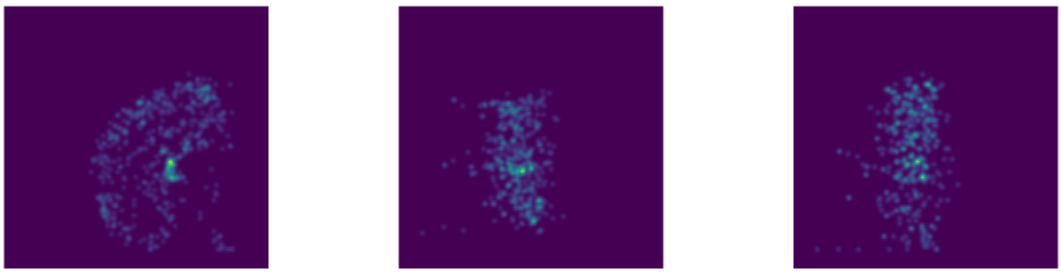
In-Silico GEX - X-Y-Z-axis Cux2



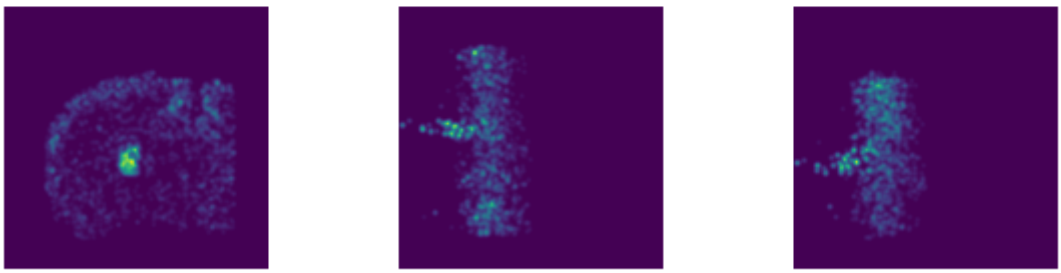

Real GEX
Lrrtm4
Real GEX (Tissue 1) - X-Y-Z-axis
Lrrtm4
Real GEX (Tissue 2) - X-Y-Z-axis
Lrrtm4
Real GEX (Tissue 3) - X-Y-Z-axis
Lrrtm4
Real GEX (Tissue 4) - X-Y-Z-axis
Lrrtm4
In-Silico GEX - X-Y-Z-axis
Lrrtm4

Real GEX Rora

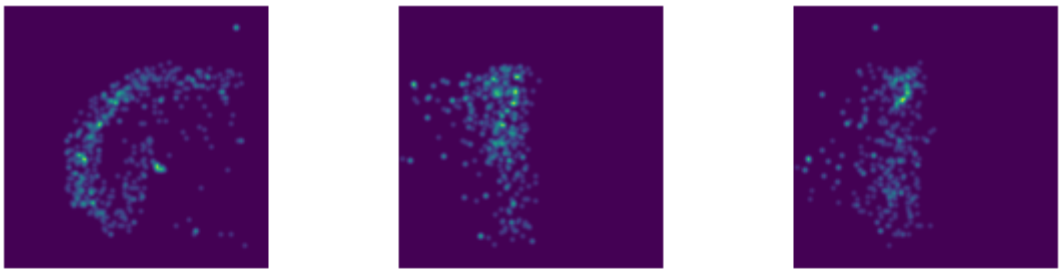
Real GEX (Tissue 1) - X-Y-Z-axis Rora



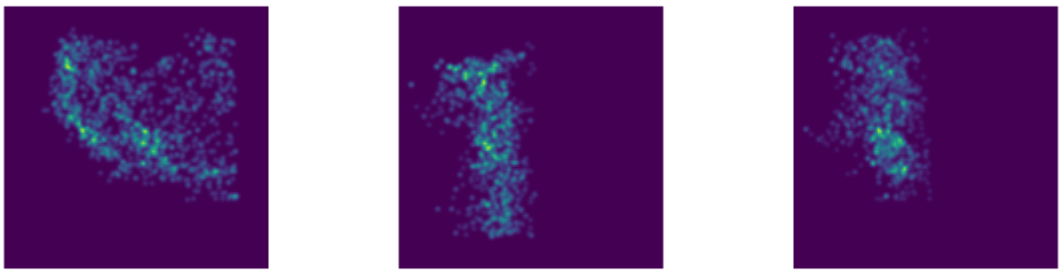
Real GEX (Tissue 2) - X-Y-Z-axis Rora



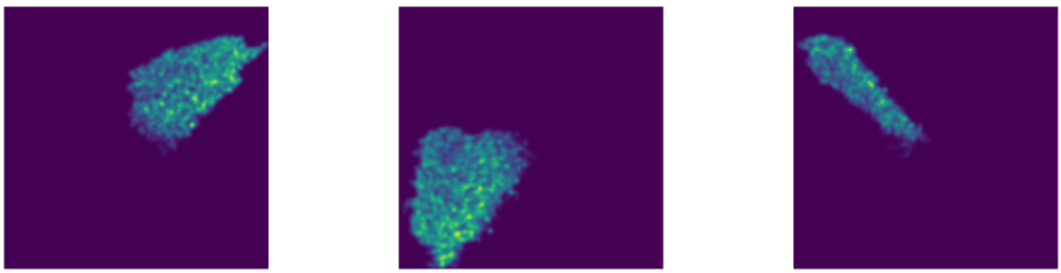
Real GEX (Tissue 3) - X-Y-Z-axis Rora



Real GEX (Tissue 4) - X-Y-Z-axis Rora

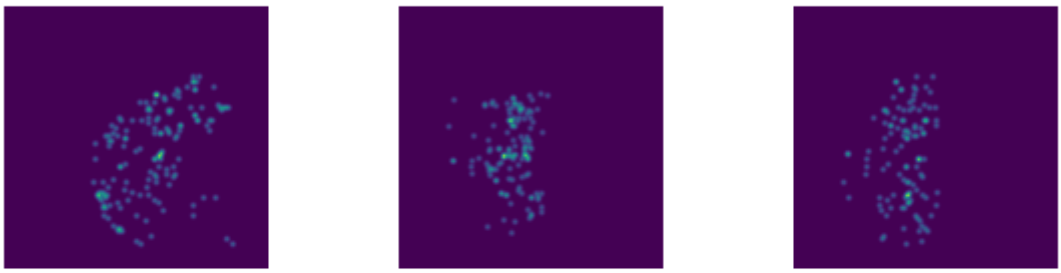


In-Silico GEX - X-Y-Z-axis Rora

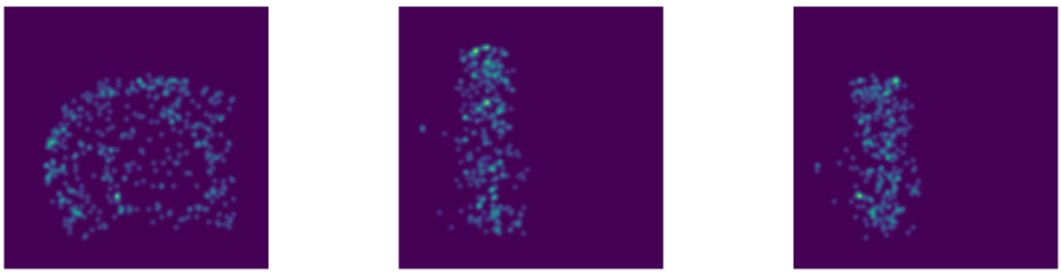


Real GEX Fat3

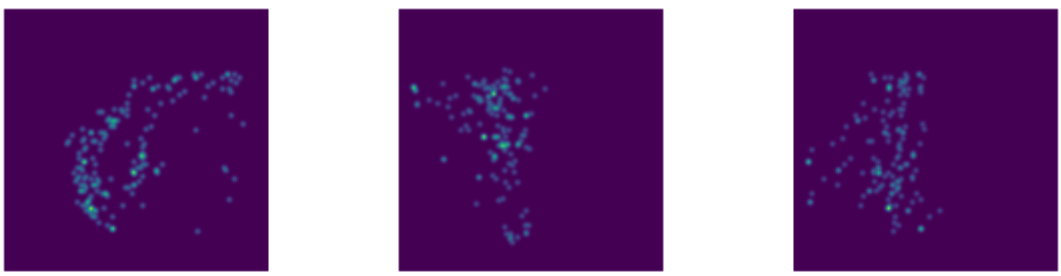
Real GEX (Tissue 1) - X-Y-Z-axis Fat3



Real GEX (Tissue 2) - X-Y-Z-axis Fat3

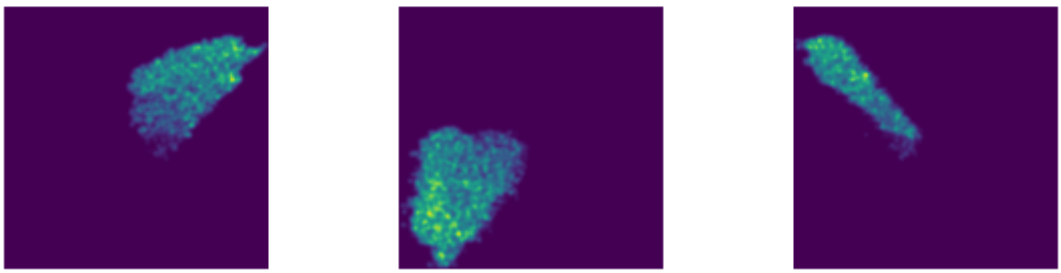


Real GEX (Tissue 3) - X-Y-Z-axis Fat3

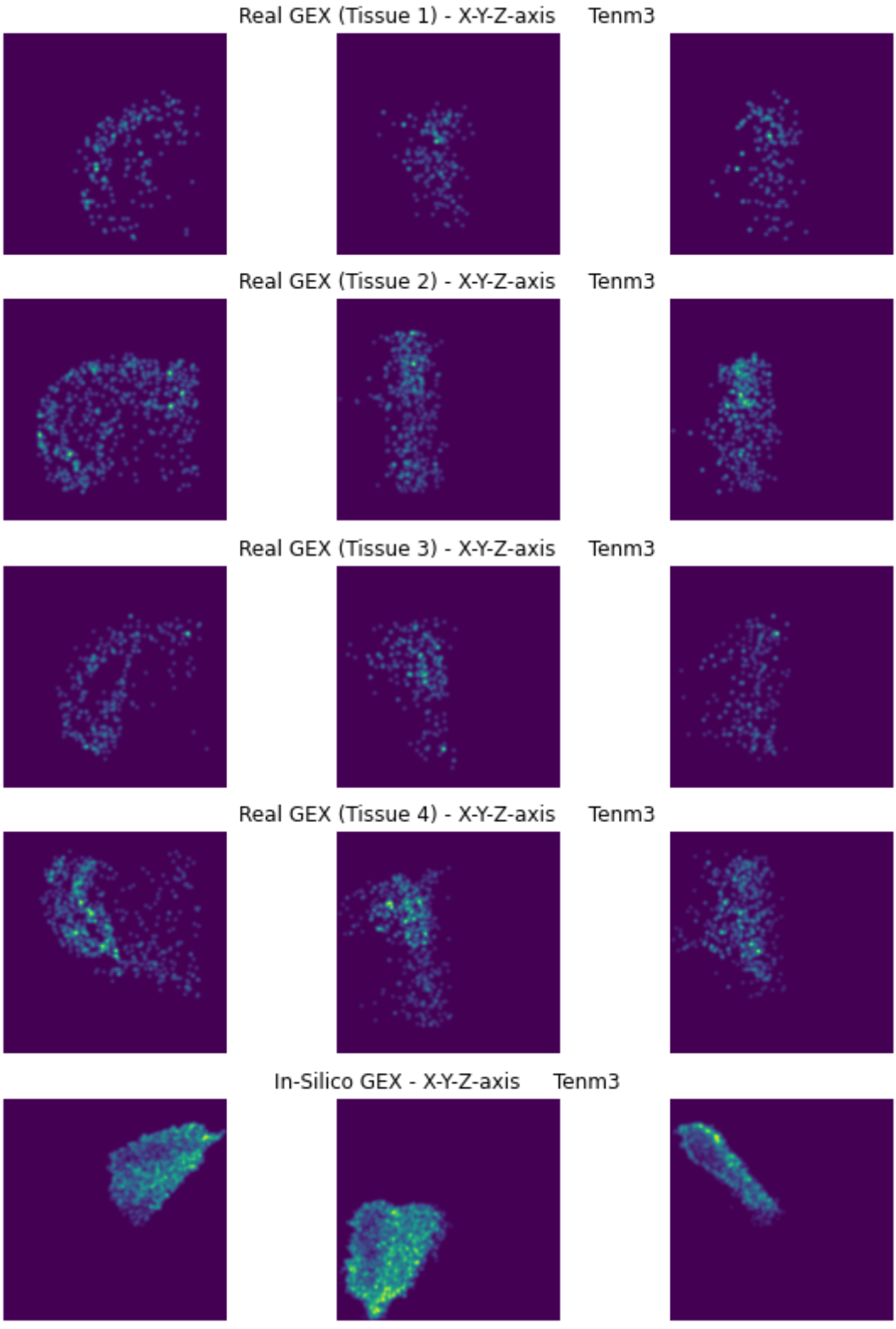


Real GEX (Tissue 4) - X-Y-Z-axis Fat3

In-Silico GEX - X-Y-Z-axis Fat3

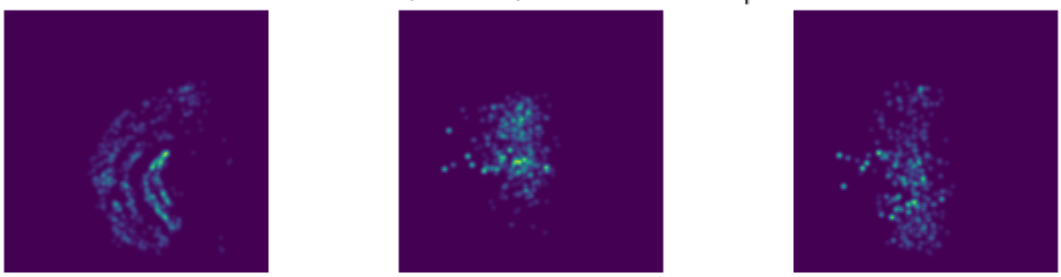


Real GEX Tenm3

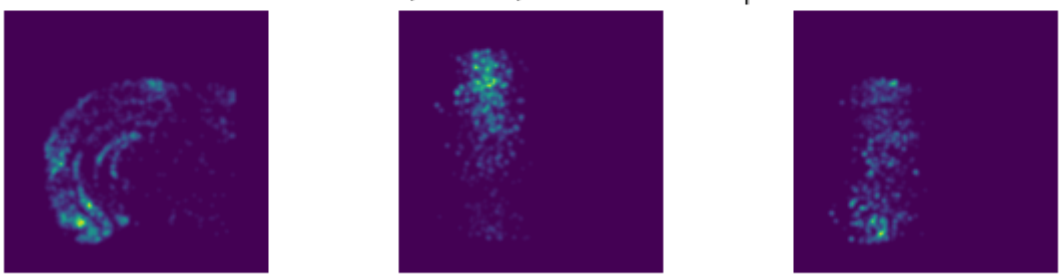


Real GEX C1ql3

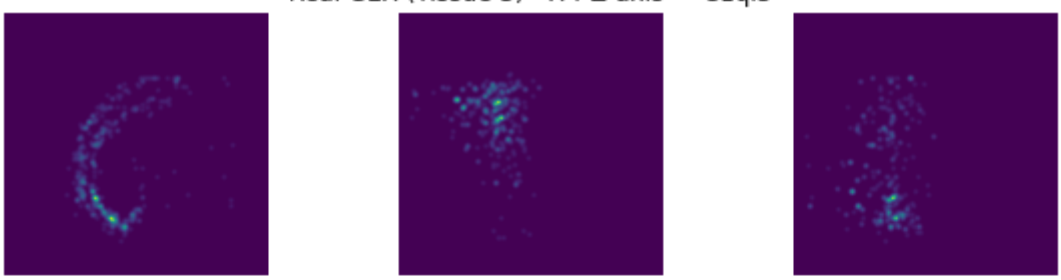
Real GEX (Tissue 1) - X-Y-Z-axis C1ql3



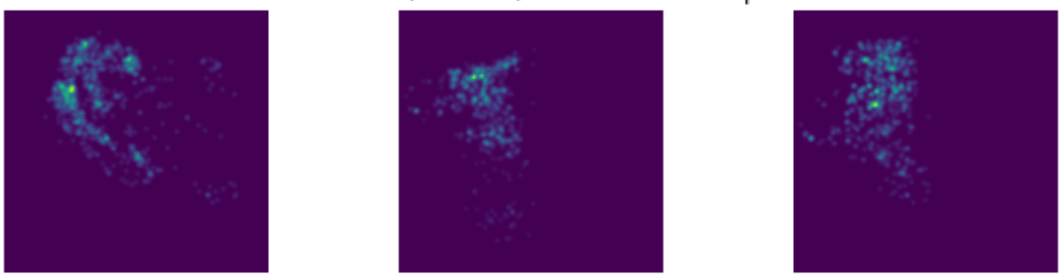
Real GEX (Tissue 2) - X-Y-Z-axis C1ql3



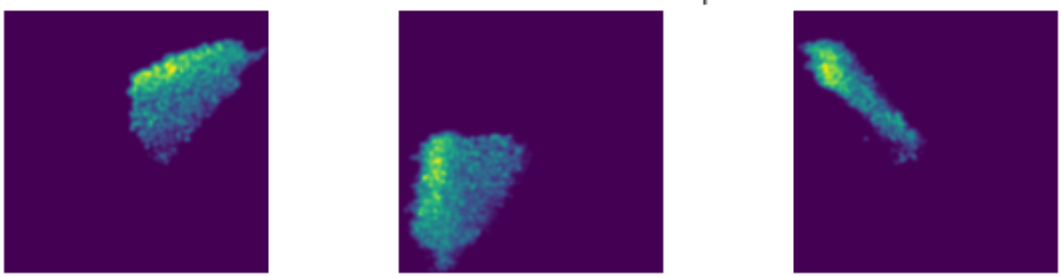
Real GEX (Tissue 3) - X-Y-Z-axis C1ql3



Real GEX (Tissue 4) - X-Y-Z-axis C1ql3

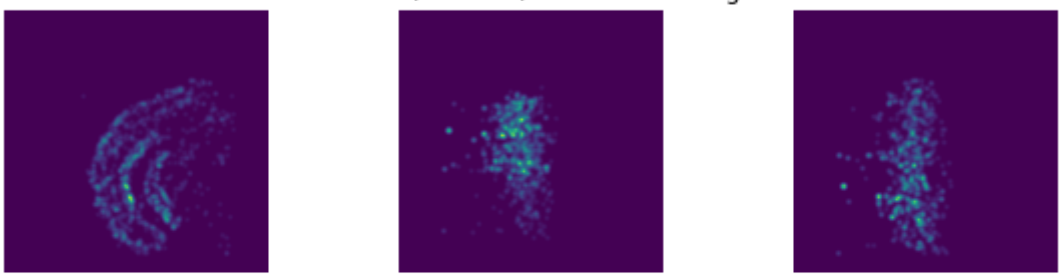


In-Silico GEX - X-Y-Z-axis C1ql3

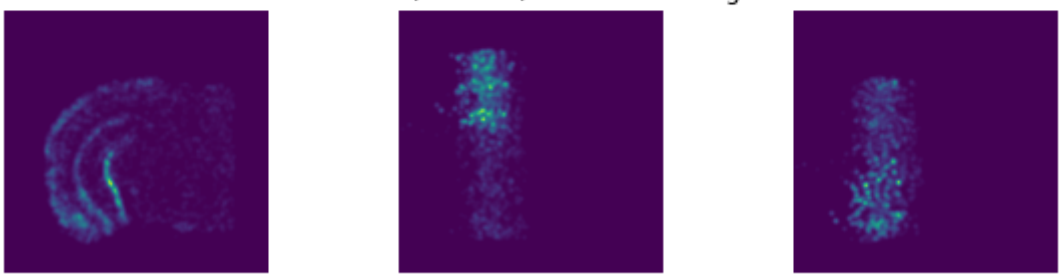


Real GEX Dgkb

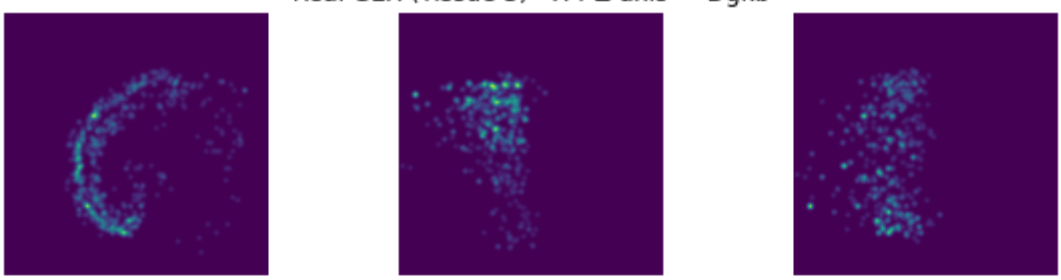
Real GEX (Tissue 1) - X-Y-Z-axis Dgkb



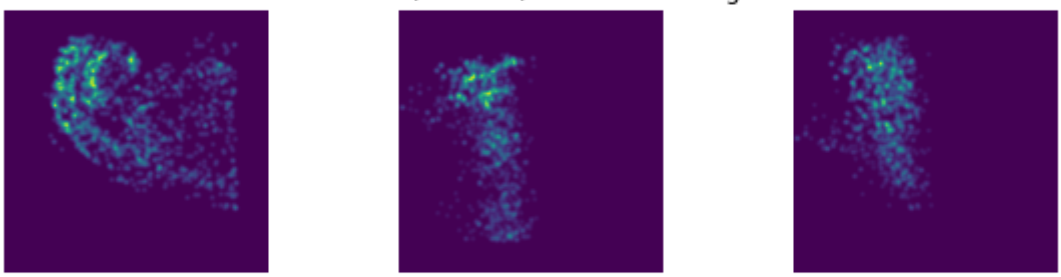
Real GEX (Tissue 2) - X-Y-Z-axis Dgkb



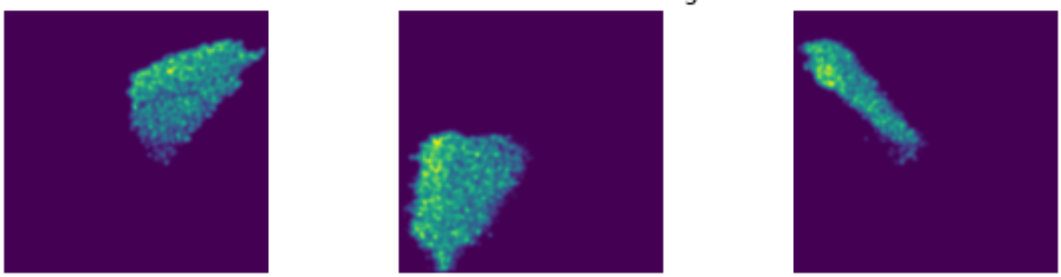
Real GEX (Tissue 3) - X-Y-Z-axis Dgkb



Real GEX (Tissue 4) - X-Y-Z-axis Dgkb

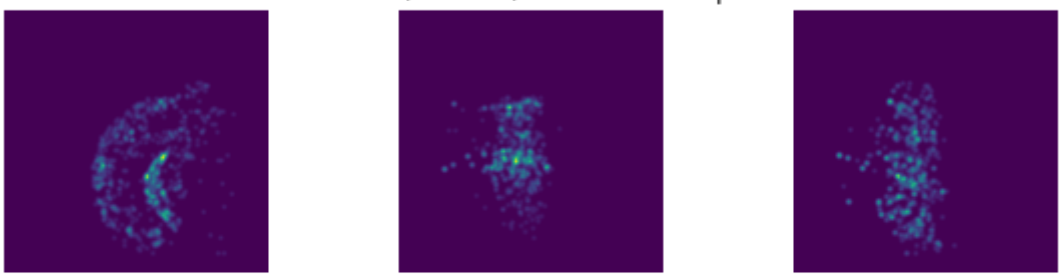


In-Silico GEX - X-Y-Z-axis Dgkb

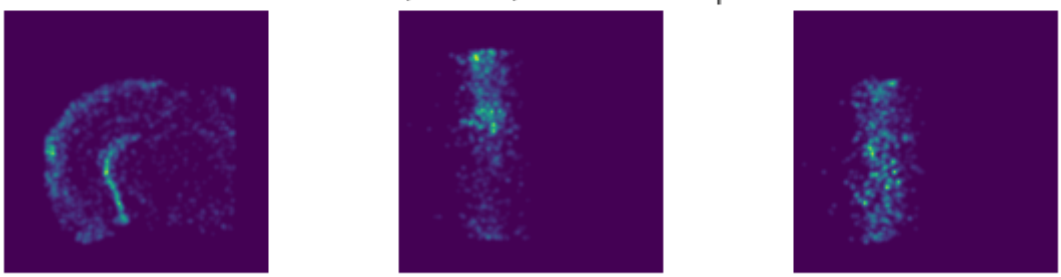


Real GEX Cpne4

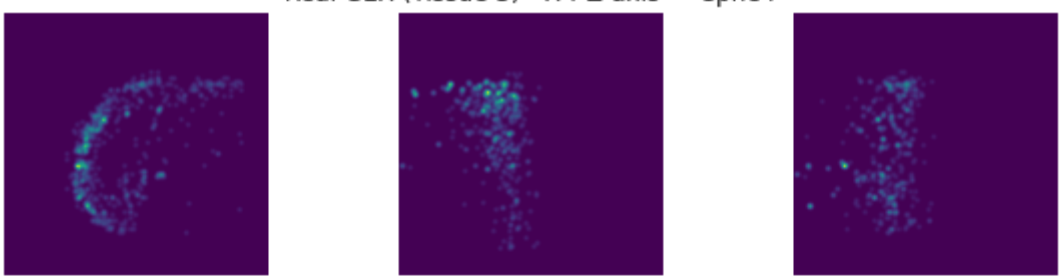
Real GEX (Tissue 1) - X-Y-Z-axis Cpne4



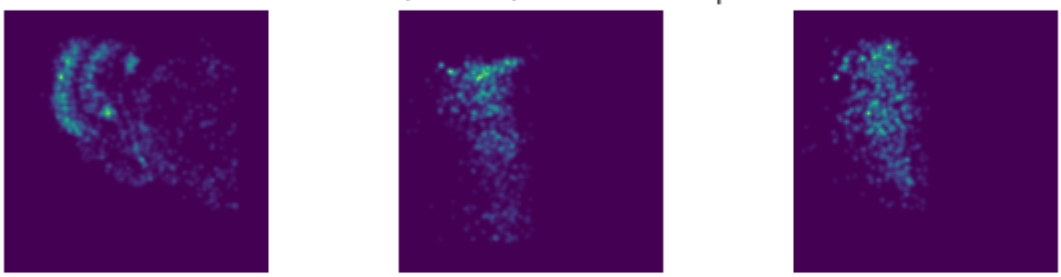
Real GEX (Tissue 2) - X-Y-Z-axis Cpne4



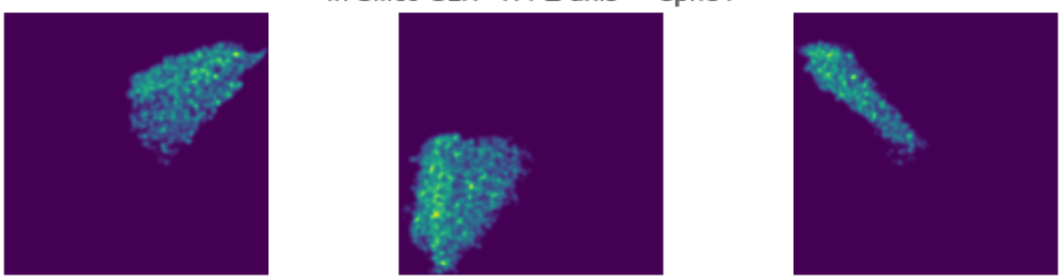
Real GEX (Tissue 3) - X-Y-Z-axis Cpne4



Real GEX (Tissue 4) - X-Y-Z-axis Cpne4

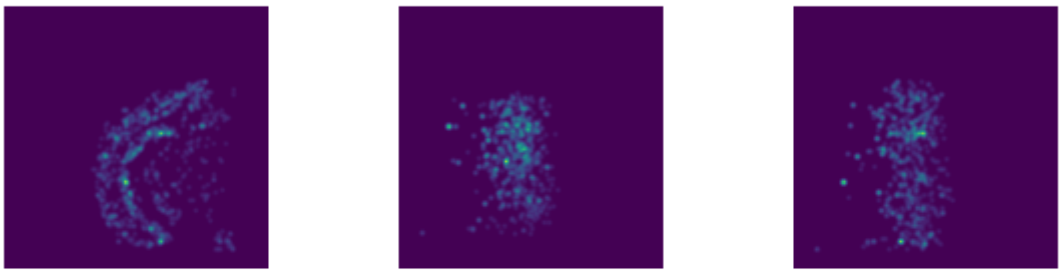


In-Silico GEX - X-Y-Z-axis Cpne4

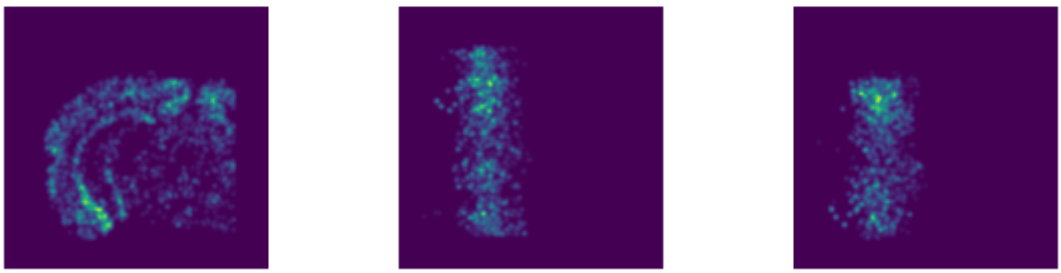


Real GEX Hcn1

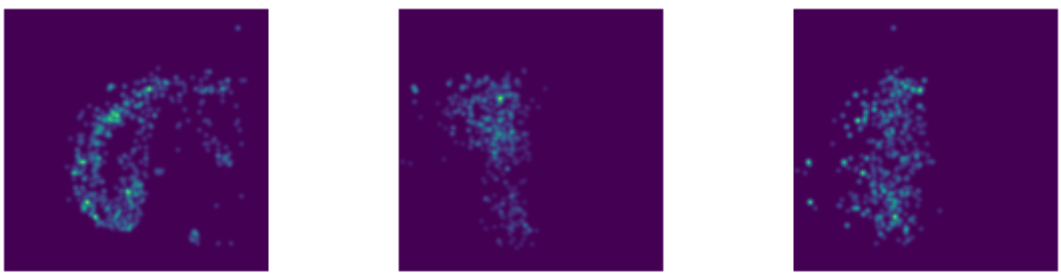
Real GEX (Tissue 1) - X-Y-Z-axis Hcn1



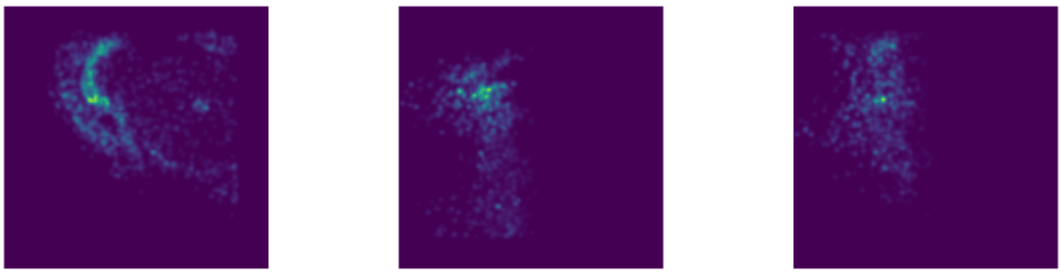
Real GEX (Tissue 2) - X-Y-Z-axis Hcn1



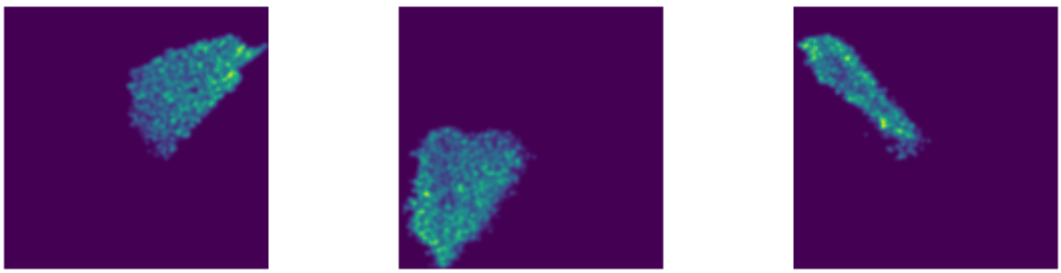
Real GEX (Tissue 3) - X-Y-Z-axis Hcn1



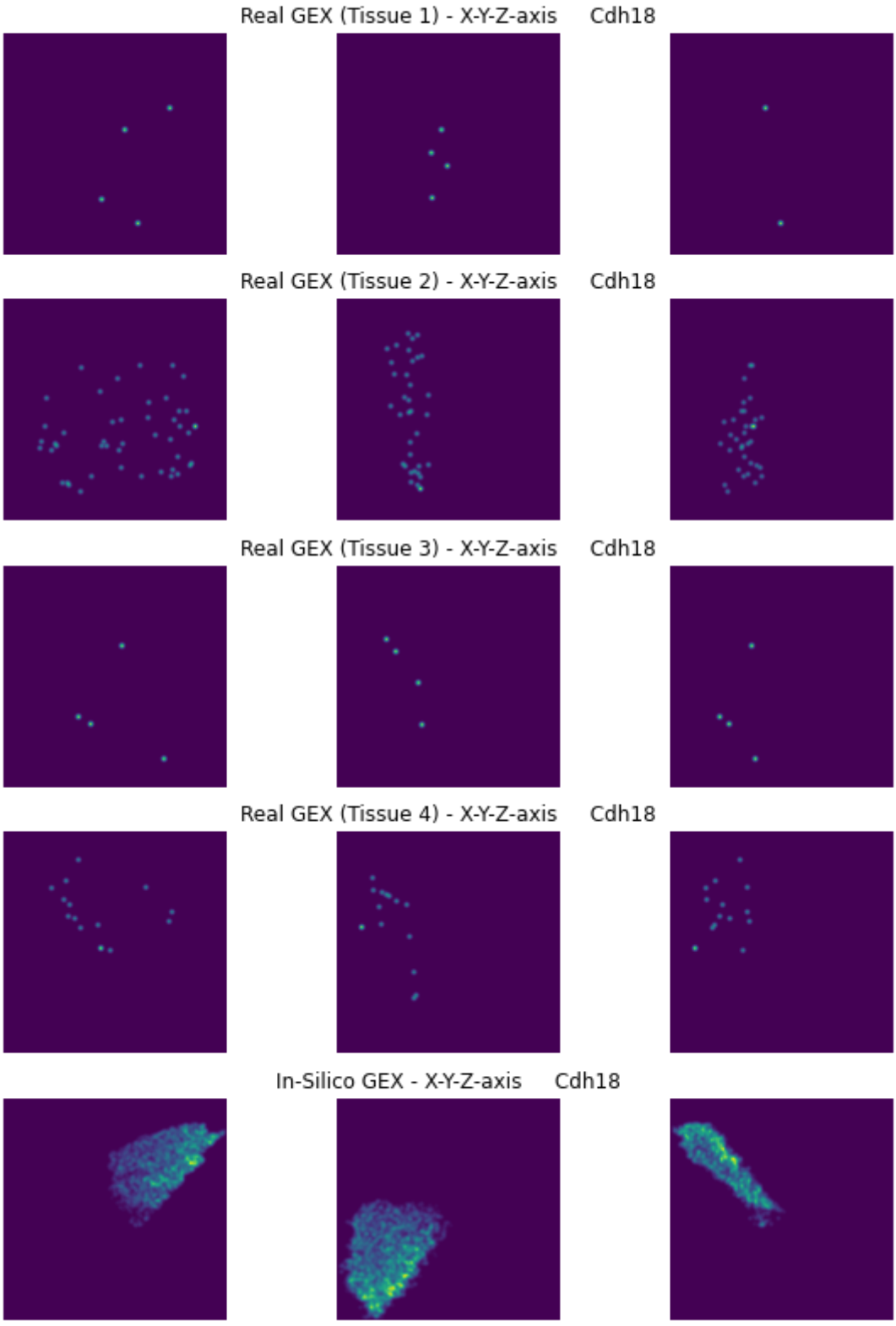
Real GEX (Tissue 4) - X-Y-Z-axis Hcn1



In-Silico GEX - X-Y-Z-axis Hcn1

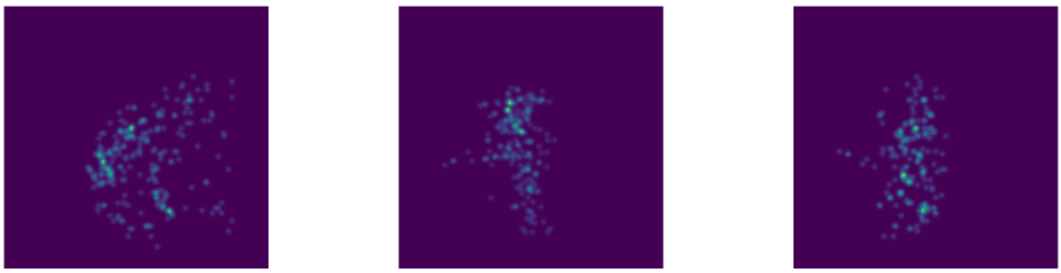


Real GEX Cdh18

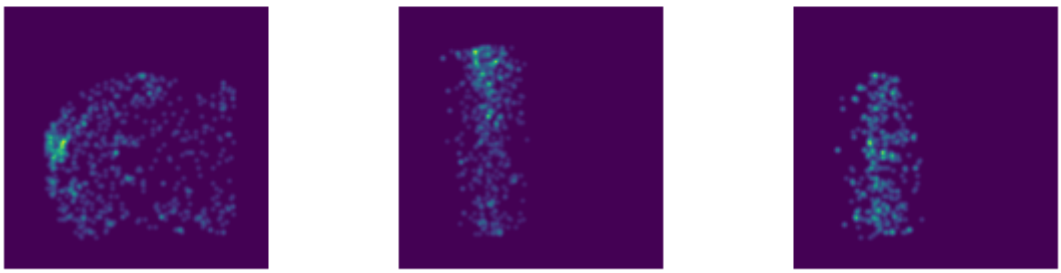


Real GEX Gnb4

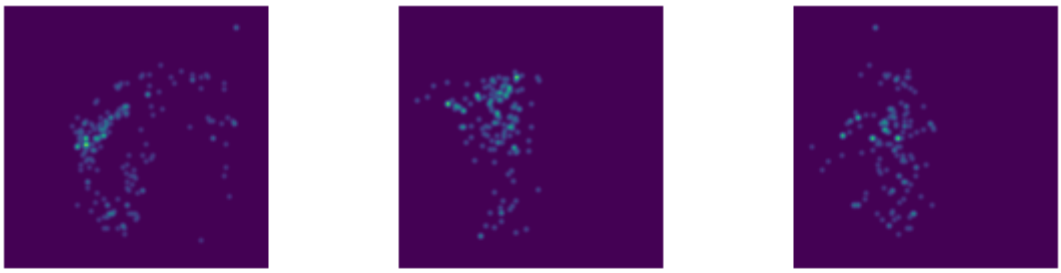
Real GEX (Tissue 1) - X-Y-Z-axis Gnb4



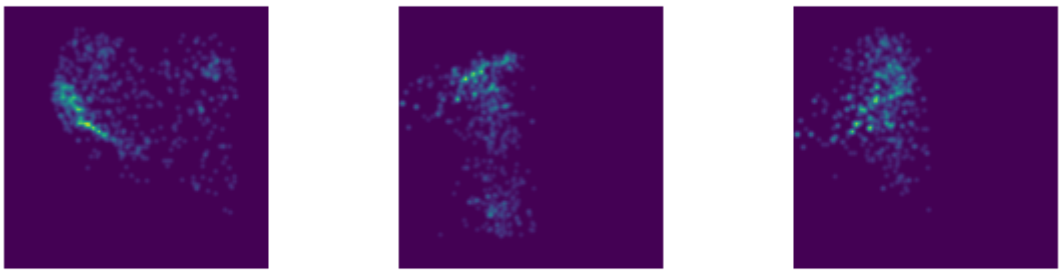
Real GEX (Tissue 2) - X-Y-Z-axis Gnb4



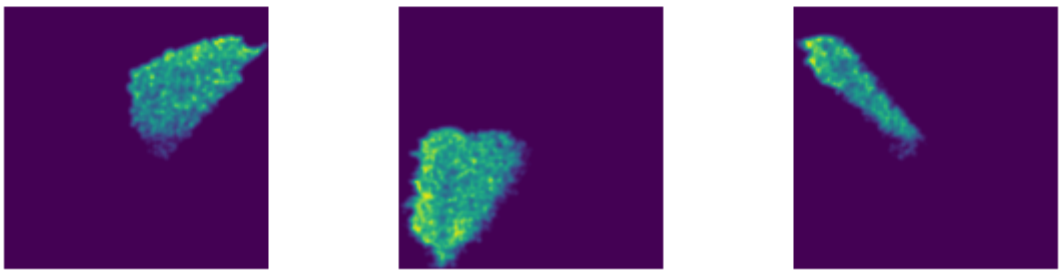
Real GEX (Tissue 3) - X-Y-Z-axis Gnb4



Real GEX (Tissue 4) - X-Y-Z-axis Gnb4

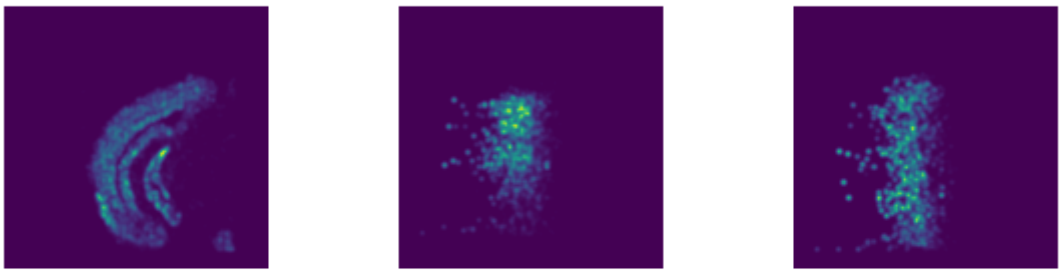


In-Silico GEX - X-Y-Z-axis Gnb4

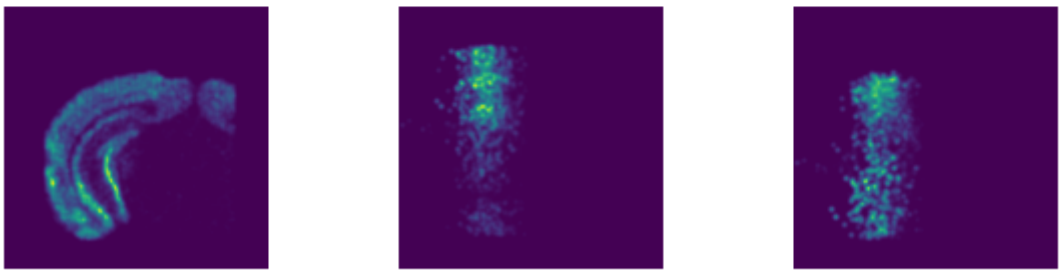


Real GEX Slc17a7

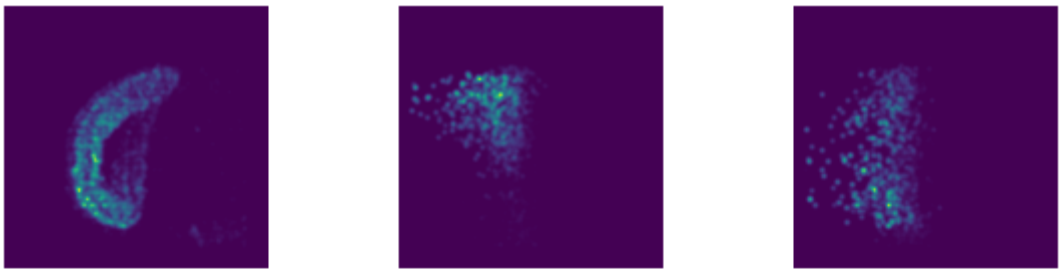
Real GEX (Tissue 1) - X-Y-Z-axis Slc17a7



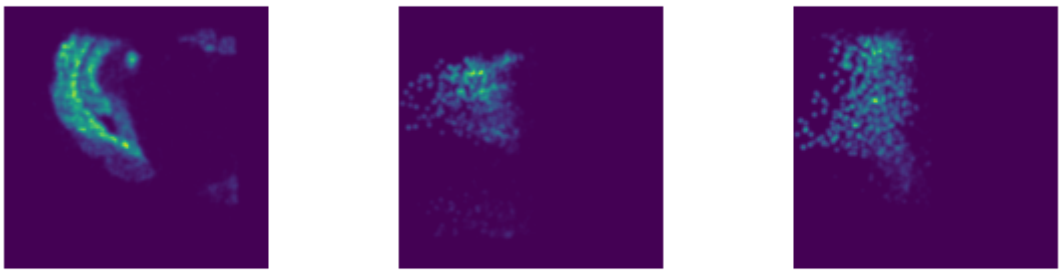
Real GEX (Tissue 2) - X-Y-Z-axis Slc17a7



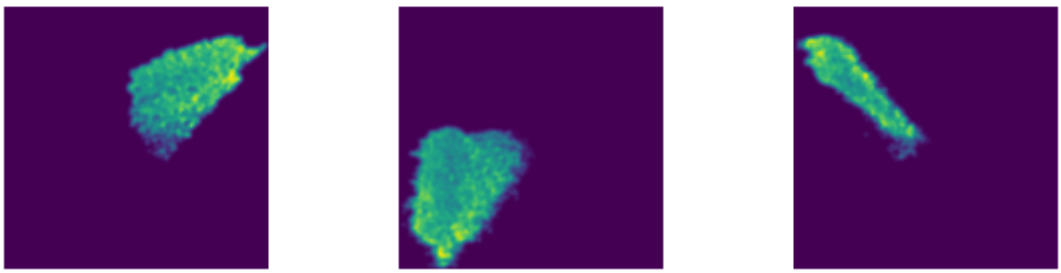
Real GEX (Tissue 3) - X-Y-Z-axis Slc17a7



Real GEX (Tissue 4) - X-Y-Z-axis Slc17a7

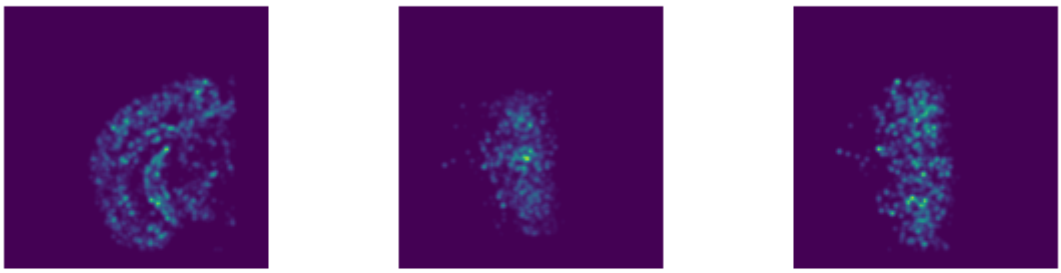


In-Silico GEX - X-Y-Z-axis Slc17a7

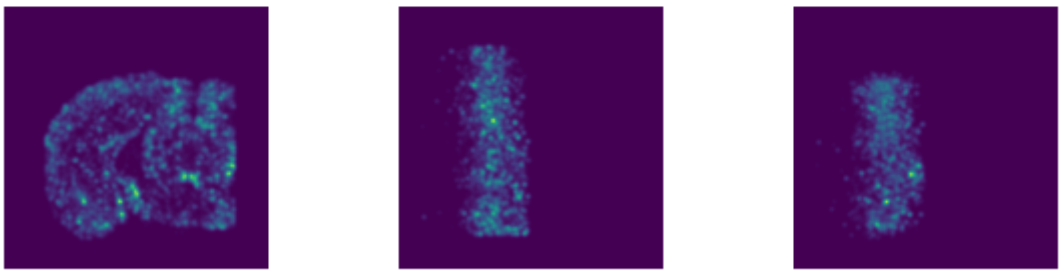


Real GEX Gad1

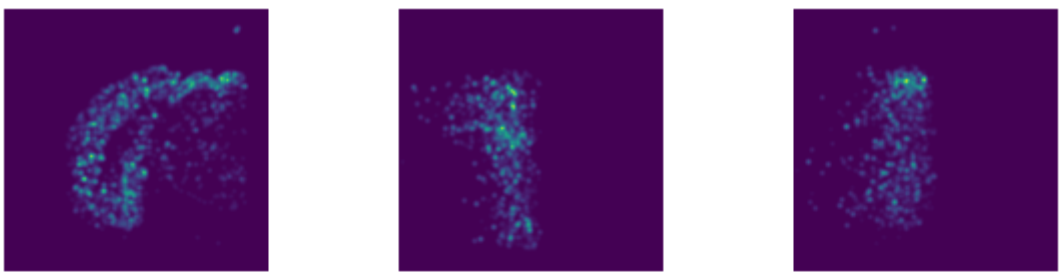
Real GEX (Tissue 1) - X-Y-Z-axis Gad1



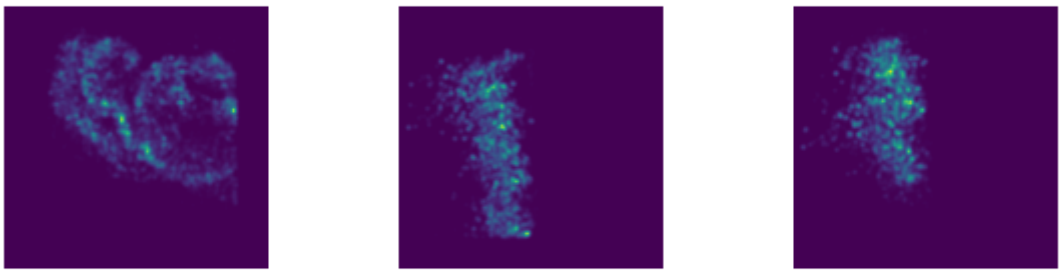
Real GEX (Tissue 2) - X-Y-Z-axis Gad1



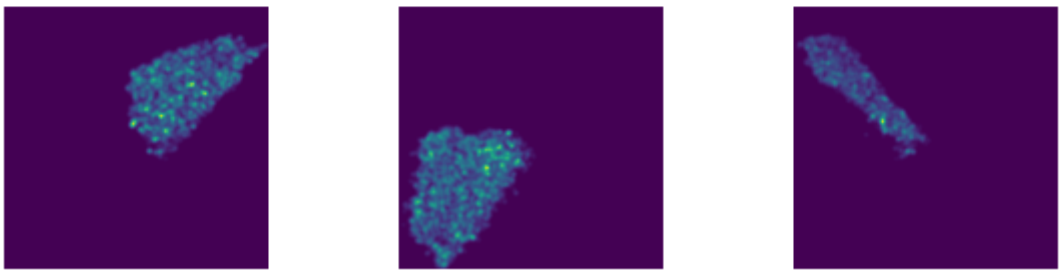
Real GEX (Tissue 3) - X-Y-Z-axis Gad1

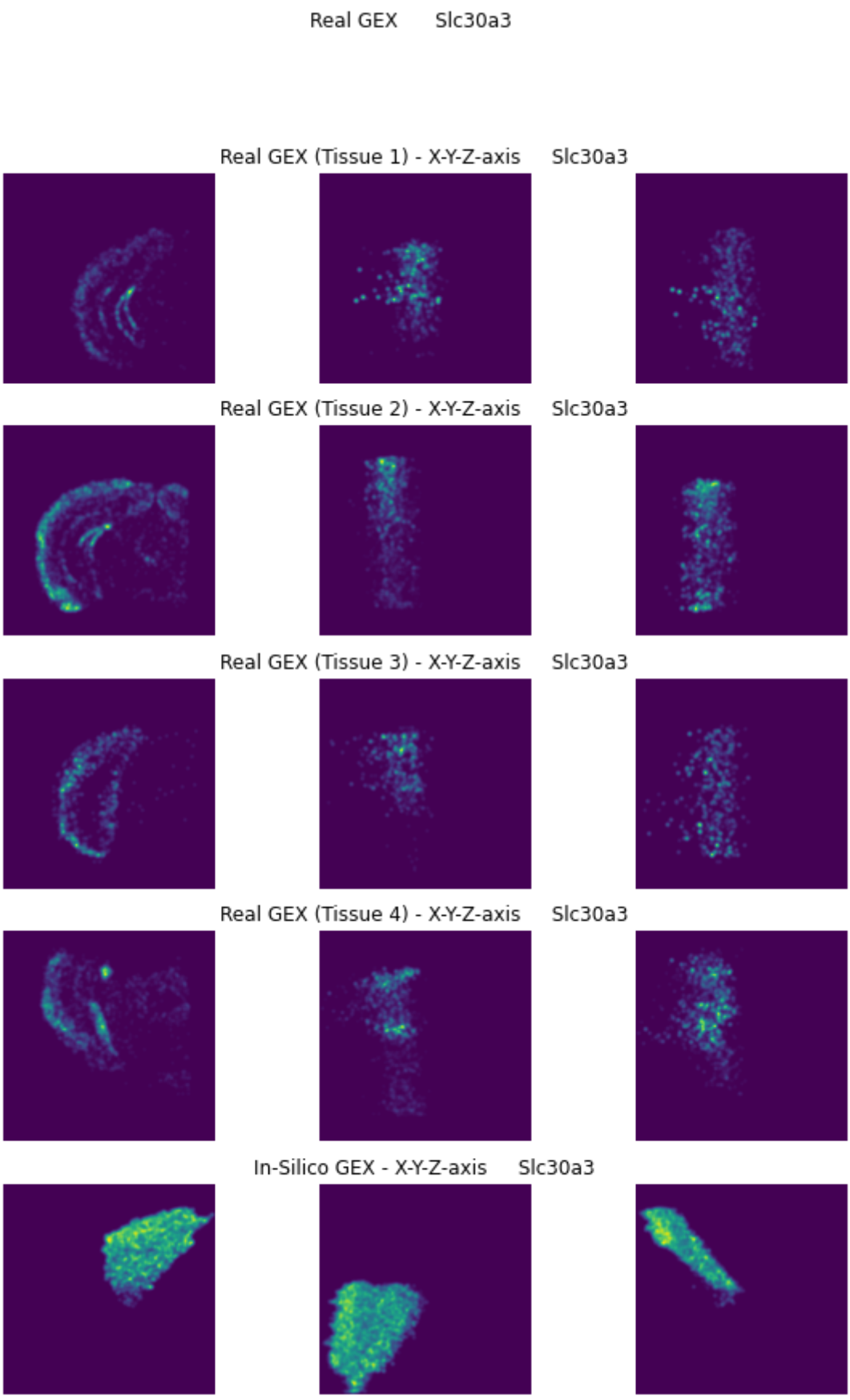


Real GEX (Tissue 4) - X-Y-Z-axis Gad1



In-Silico GEX - X-Y-Z-axis Gad1




Real GEX Slc30a3
Real GEX (Tissue 1) - X-Y-Z-axis Slc30a3
Real GEX (Tissue 2) - X-Y-Z-axis Slc30a3
Real GEX (Tissue 3) - X-Y-Z-axis Slc30a3
Real GEX (Tissue 4) - X-Y-Z-axis Slc30a3
In-Silico GEX - X-Y-Z-axis Slc30a3

In [ ]: