ISS Live Script

This Matlab Live Scripts summarizes the functions developed in order to analyse the spatial profiles derived from ISS. However, they can be applied to other spatial technologies, since the only requirement they present is the spatial distribution of genes in 2D, together with a DAPI background image representing the nuclei.

Loading the data

The path to the file containing the spatial expression of our genes of interest is saved in decoded_file. The background used, in our case DAPI, is saved in **image**. We extract the coordinates from the decoded_file using the function **ISS_getspots** and we save it on a SPOTS object. This SPOTS object contains the main information regarding Gene expression, its location, background image and output location.

```
decoded_file = 'G:\DIPG pciseq\Sample2\QT_0.4_details_noNNNN_goodS2.csv';
image = 'G:\DIPG pciseq\Sample2\Base_2_aligned-1.tif';    % important for size
scale=1;
%% do not modify
% load
SPOTS=ISS_getspots(decoded_file);
SPOTS.image=image;
SPOTS.scale=scale;
SPOTS.output_directory='G:\outputfolder';
mkdir(SPOTS.output_directory);
```

Gene Density maps

```
%ISS_GeneDensity plots the density of 1 gene
genes_density = {'SIRT2'};
bandwid = 200;   % in original scale
ISS_GeneDensity(SPOTS, genes_density, bandwid)
```

Overay

```
%ISS_OverlayTwoDensities overlays 2 densities

genes_density = {'SIRT2', 'WASF1'};       % first in red, second in cyan
ISS_OverlayTwoDensities(SPOTS,genes_density,bandwid);
```

Dissimilarity test

```
% ISS_DissimilarityTest test for dissimilarity
    %varargin1 is minimum number of genes per cell
    % varargin2 is hexbin size
 ISS_DissimilarityTest(SPOTS,20);
```

Find Neighbors

```
 ISS_FindNeighbors_screen_randomization(SPOTS)
```

Search AND regions

```
ANDgenes= {'SIRT2', 'WASF1'};
ISS_SearchANDRegions(SPOTS,ANDgenes);
```

Search for regions with Similarity

```
transcripts={'SIRT2', 'WASF1'};

ISS_A_B_neighbors(SPOTS,transcripts);
```

Segmentation techniques

Segmenting cells

```
percent=95;
[CellMap,blobs,XY]=ISS_DAPI_segmentation(SPOTS,percent);
```

```
%XY=fliplr(vertcat(regionprops(CellMap).Centroid));

ISS_plotCellReads(SPOTS,CellMap)

[TOTAL,EXPRESSIONMAT,CELS]= ISS_assign_gene_to_cell(SPOTS,CellMap,XY);
```

Creating overlapping means

We include on 2nd position distance and on 3rd radius

```
[OUTPUT] =ISS_OverlappingBins(SPOTS,800,800)
```

RGB-Tsne representations

Tsne representation

```
hexbin_size=40;
minimum_expression=70
```

```
minimum_expression = 70
```

```
ISS_tsneRGB(OUTPUT,SPOTS,minimum_expression); %EXPRESSIONMAT, SPOTS
```

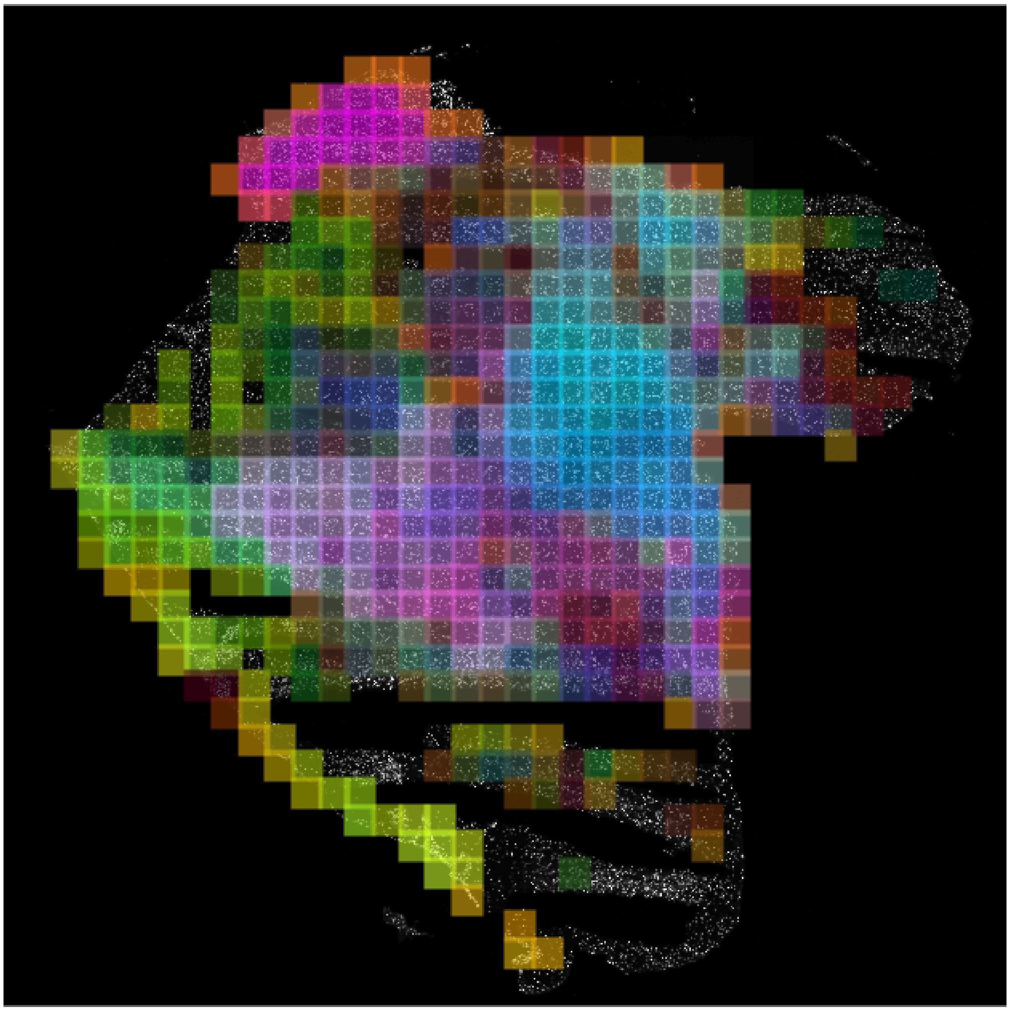
```
genes = 27×1 cell array
'AGT'
'APOE'
'AQP4'
'ARC'
'ASCL1'
'ATP1A2'
'BCAS1'
'BMPER'
'CCND1'
'EDNRB'
      ⋮

|==============================================|
|   ITER    | KL DIVERGENCE    | NORM GRAD USING |
|           | FUN VALUE USING  | EXAGGERATED DIST|
|           | EXAGGERATED DIST| OF X             |
|           | OF X             |                 |
|==============================================|
|        20 |    1.359263e+01 |    1.351119e-02 |
|        40 |    1.273087e+01 |    1.858484e-02 |
|        60 |    1.308312e+01 |    1.721224e-02 |
|        80 |    1.306211e+01 |    1.855929e-02 |

|==============================================|
|   ITER    |  KL DIVERGENCE  |     NORM GRAD    |
|           |    FUN VALUE    |                  |
|==============================================|
|       100 |    1.916487e+00 |    4.892057e-03 |
|       120 |    5.405834e-01 |    1.935398e-03 |
|       140 |    4.856481e-01 |    3.137617e-04 |
|       160 |    4.697897e-01 |    1.910719e-04 |
|      180 |    4.627087e-01 |    2.167107e-04 |
|      200 |    4.590339e-01 |    9.319219e-05 |
|      220 |    4.559642e-01 |    9.227559e-05 |
|      240 |    4.541603e-01 |    1.033573e-04 |
|      260 |    4.532877e-01 |    2.037421e-04 |
|      280 |    4.512334e-01 |    1.588832e-04 |
|      300 |    4.500712e-01 |    1.434493e-04 |
|      320 |    4.496246e-01 |    1.560741e-04 |
|      340 |    4.492876e-01 |    1.343547e-04 |
|      360 |    4.483117e-01 |    1.383807e-04 |
|      380 |    4.479211e-01 |    1.659089e-04 |
|      400 |    4.469336e-01 |    1.420770e-04 |

|==============================================|
|   ITER    |  KL DIVERGENCE  |    NORM GRAD     |
|          |    FUN VALUE    |                 |
|==============================================|
|      420 |    4.473327e-01 |    9.139782e-05 |
|      440 |    4.469998e-01 |    1.146710e-04 |
|      460 |    4.462166e-01 |    1.348416e-04 |
|      480 |    4.460838e-01 |    1.434524e-04 |
|      500 |    4.456917e-01 |    1.129301e-04 |
|      520 |    4.453566e-01 |    1.370864e-04 |
|      540 |    4.451655e-01 |    1.679850e-04 |
|      560 |    4.450961e-01 |    1.800659e-04 |
|      580 |    4.449281e-01 |    1.419631e-04 |
|      600 |    4.447933e-01 |    9.635951e-05 |
|      620 |    4.449296e-01 |    6.715089e-05 |
|      640 |    4.446256e-01 |    1.650494e-04 |
|      660 |    4.446179e-01 |    1.067745e-04 |
|      680 |    4.448808e-01 |    9.199407e-05 |
|      700 |    4.450608e-01 |    1.844477e-04 |
|      720 |    4.448551e-01 |    1.448941e-04 |
|      740 |    4.445809e-01 |    1.701401e-04 |
|      760 |    4.447250e-01 |    1.248392e-04 |
|      780 |    4.444820e-01 |    9.698355e-05 |
|      800 |    4.440157e-01 |    8.772584e-05 |

|==============================================|
|   ITER    |  KL DIVERGENCE  |    NORM GRAD     |
|          |    FUN VALUE    |                 |
|==============================================|
|      820 |    4.439672e-01 |    1.070800e-04 |
|      840 |    4.437035e-01 |    1.081337e-04 |
|      860 |    4.436578e-01 |    6.759112e-05 |
|      880 |    4.434808e-01 |    1.233432e-04 |
|      900 |    4.435083e-01 |    7.233830e-05 |
|      920 |    4.434568e-01 |    8.208956e-05 |
|      940 |    4.433521e-01 |    8.468236e-05 |
|      960 |    4.434418e-01 |    1.325100e-04 |
|      980 |    4.434189e-01 |    1.632068e-04 |
|     1000 |    4.432658e-01 |    1.076185e-04 |
```



Working with expression matrix

This includes segmented data and binned data

Clustering data

```
CELLSHIERARCHICAL= ISS_hierarchical_clustering(OUTPUT,SPOTS,6)
```

PCISEQ

This part of PCIseq allows you to select

```
BigDapiImage = imread(SPOTS.image);
figure(707);
imagesc(min(BigDapiImage,prctile(BigDapiImage(:),98)));
colormap(bone);
set(gca, 'YDir', 'normal');
[~, xpoly, ypoly] = roipoly;
```

Remember to load o to path

```
SPOTS.CellCallRegionYX=[ypoly,xpoly];




load E:\pciSeq\Data\AllSections\gSetCA1all.mat;
%CLUSTERED DATA ~ human cortex
gSet.GeneExp=readmatrix('H:\hm_brain_SBH_small\cortex_scRNAseq\Human_cortex_scRNAseq_dataset.cs
nam=readcell('H:\hm_brain_SBH_small\cortex_scRNAseq\Human_cortex_scRNAseq_dataset.csv');
namclust=readcell('H:\hm_brain_SBH_small\cortex_scRNAseq\Human_cortex_scRNAseq_dataset.csv');
gSet.GeneExp=gSet.GeneExp(:,2:end);
gSet.GeneExp=gSet.GeneExp';
gSet.GeneName=nam(2:end,1);
gSet.Class=namclust(1,2:end);
gSet.CellName=gSet.Class(1:size(gSet.Class,2));
gSet.nGenes=size(gSet.GeneExp,2);
gSet.nCells=size(gSet.GeneExp,1);
```

This is gSet for DIPG

```
load E:\pciSeq\Data\AllSections\gSetCA1all.mat;
gSet.GeneExp=readmatrix('G:\DIPG pciseq\expressionmatrix_selected.csv');
gSet.GeneExp=gSet.GeneExp(:,3:end);
nam=readcell('G:\DIPG pciseq\expressionmatrix_selected.csv');
gSet.GeneName=nam(2:end,2);
namclust=readcell('G:\DIPG pciseq\metadata_selected.csv');
gSet.Class=namclust(2:end,4);
gSet.CellName=gSet.CellName(1:size(gSet.Class,1));
gSet.nGenes=size(gSet.GeneExp,2);
gSet.nCells=size(gSet.GeneExp,1);
%THIS LINE IS FOR CONSIDERING CELL TYPE AS INDIVIDUAL CLASSES
%gSet.Class=gSet.CellName;
```

Running PCIseq itself

```
 CELLS = ISS_pciseq(SPOTS,CellMap,gSet)
gscatter(CELLS.pos(:,2),CELLS.pos(:,1),CELLS.name(:,1:end-1)')
```

Selection based on CELS

```
SELECTED.clusters=c;
SELECTED.pos=CELLS.pos(CELS,:);
SELECTED.name=CELLS.name(:,CELS);

[celltypecor,n,n,labels]=crosstab(SELECTED.name,SELECTED.clusters);

figure
XC=unique(SELECTED.clusters);
YC=unique(SELECTED.name);
h = heatmap(labels(:,2),labels(2:6,1),celltypecor(2:6,:))
```

Heatmap on PCISEQ

Plotting a heatmap of differentially expressed clusters

```
ISS_heatmap_cluster(OUTPUT,CELLSHIERARCHICAL.name')
```

```
1

2

3

4

5

6
```

```
%ISS_heatmap_cluster(EXPRESSIONMAT,SELECTED.name')
```

ROI selection

```
output_directory='G:\Tools-master\Visualization tools\Analize Region of Interests\ROI';
roi_number=2;
%LAST TWO TERMS CAN BE IN OR OUR DEPENDING IF YOU WANT TO SELECT BY DENSITY
%(adding 'GPR17',100)
%OR DAPI
[INROI,ROI_borders] = ISS_ROI_draw_onImage(SPOTS,output_directory,roi_number);
```

Now we select ROI based on density

-THIS IS SOMETHING THAT CAN BE ADDED

GRADIENT ANALYSIS

We define a gradient expression based on a ROI defined by ourselves

```
limits=800;
bandwidth=100;
ISS_gradients(CELLS,limit,bandwidth)
```