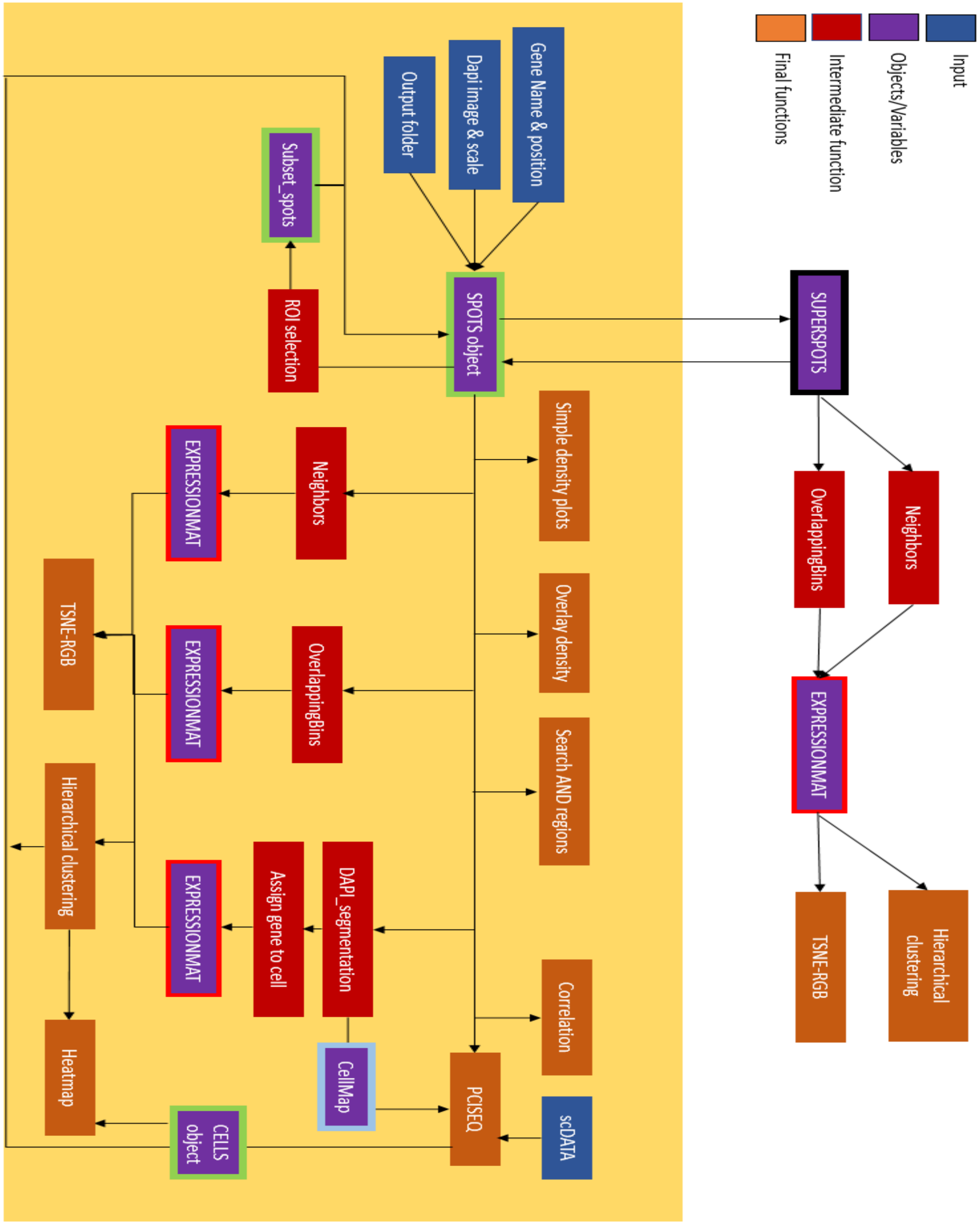
ISS Analysis of 1 sample

This Matlab Live Script summarizes the functions developed in order to analyse the spatially resolved transcriptomic profiles based on dots. Therefore, they can be applied to different 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 / centroids. The main structure of this

This specific notebook analyse a Difuse Intrincic Pons Glioma (DIPG) sample, as an example of analysis tools.



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.

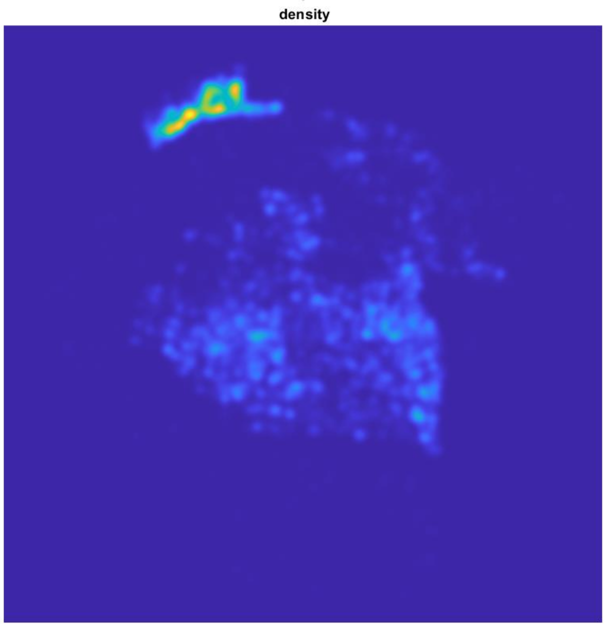
```
decoded_file = 'G:\DIPG pciseq\Sample2\QT_0.4_details_noNNNN_goodS2.csv';
image = 'G:\DIPG pciseq\Sample2\Base_2_aligned-1.tif';    % important for size
scale=1;
%% do not modify
% load
SPOTS=ISS_getspots(decoded_file);
SPOTS.image=image;
SPOTS.scale=scale;
SPOTS.output_directory='G:\outputfolder';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

```
Warning: Directory already exists.
```

Gene Density maps

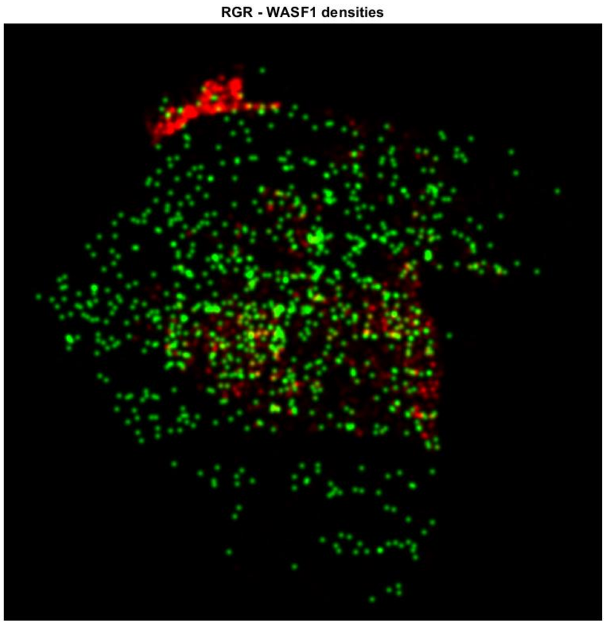
The function ISS_GeneDensity creates a density map for the desired gene. We need to specify a Bandwidth for the density map.

```
genes_density = {'RGR'};
bandwid = 200;    % in original scale
ISS_GeneDensity(SPOTS, genes_density, bandwid)
```



ISS_OverlayTwoDensities generates a 2D density map with the density of 2 genes. Two maps are actually generated, one with Smooth kernel and another one without.

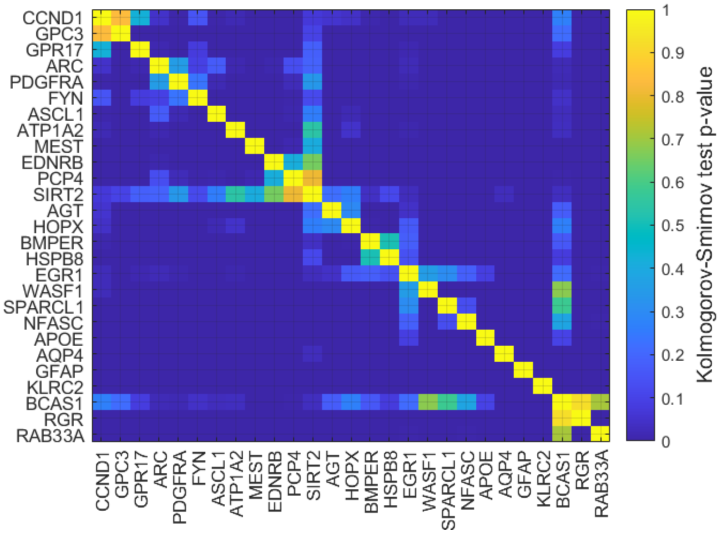
```
%ISS_OverlayTwoDensities overlays 2 densities
genes_density = {'RGR', 'WASF1'};       % first in red, second in cyan
bandwid=100;
ISS_OverlayTwoDensities(SPOTS,genes_density,bandwid);
```



Dissimilarity test

Dissimilarity test search for coexpression of genes, taking into account a different number of counts on each case. We need to specify the minumum number of genes per bin/cell, together with

```
% ISS_DissimilarityTest test for dissimilarity
    %varargin1 is minimum number of genes per cell
    % varargin2 is hexbin size
 ISS_DissimilarityTest(SPOTS,1,1000);
```



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);
```

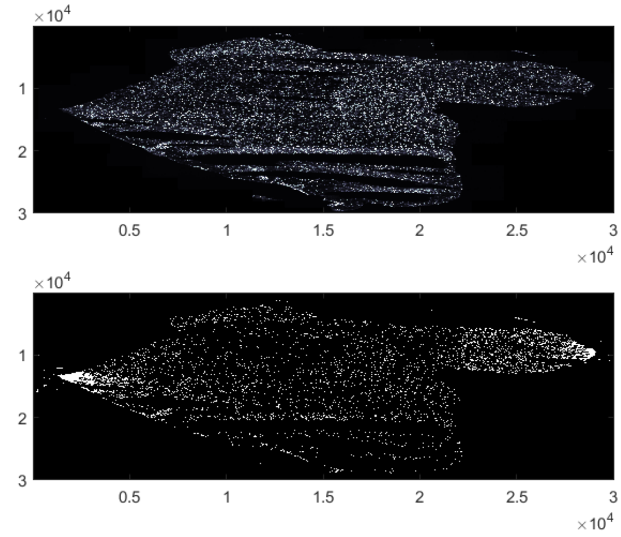
Segmentation techniques

Our data can be also analysed considering the expression of genes in a specific region and comparing the expression between these regions. This regions can be individual cells, defined by the presence of nuclei/membrane, or different bins, distributed homogeneously across the tissue without any prior knowledge.

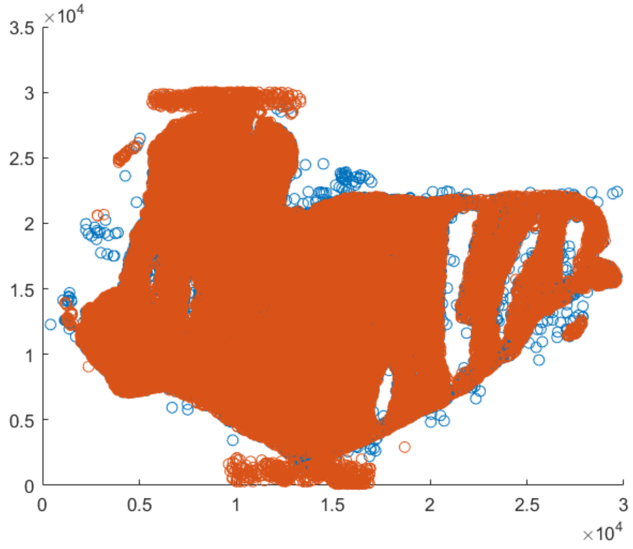
Segmenting cells via DAPI

In order to analyse the expression of every individual genes, we first segment a DAPI image and, from this segmentation, we define the cells boundries and the expression within each cell.

```
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

As mentioned, we also do capture the expression of different regions in the tissue by binning it and capturing the expression profile. In this case *OverlappingBins* generates bins that overlap between them. In this case, we can specify on 2nd position distance between spots in pixels and on 3rd position the radius of the bin

```
[OVERLAPPING] =ISS_OverlappingBins(SPOTS,200,800)
```

```
i = 1
OVERLAPPING = struct with fields:
           exp: [13546×27 double]
           loc: [13546×2 double]
      genename: {27×1 cell}
    hexbinsize: 60
```

RGB-Tsne representations

Tsne representation

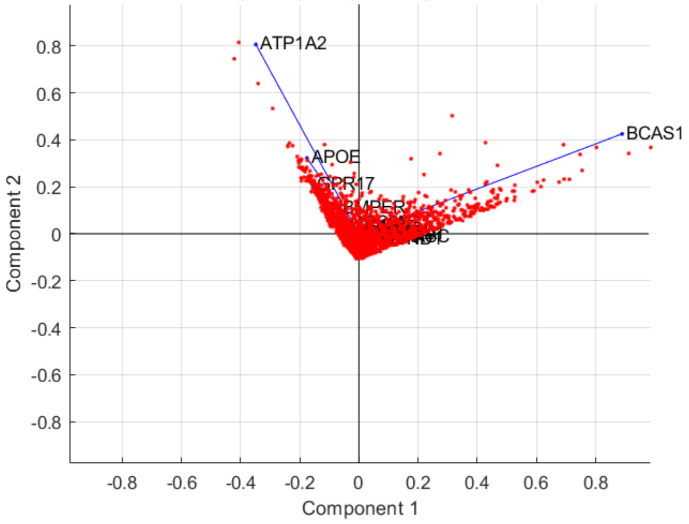
In order to represent the heterogeneity in a tissue, we use binned or segmented data to perform 3D tSNE, representing each of the dimensions in RGB scale. Therefore, we are able to represent differentially expressed regions. We can select the genes going into the analysis

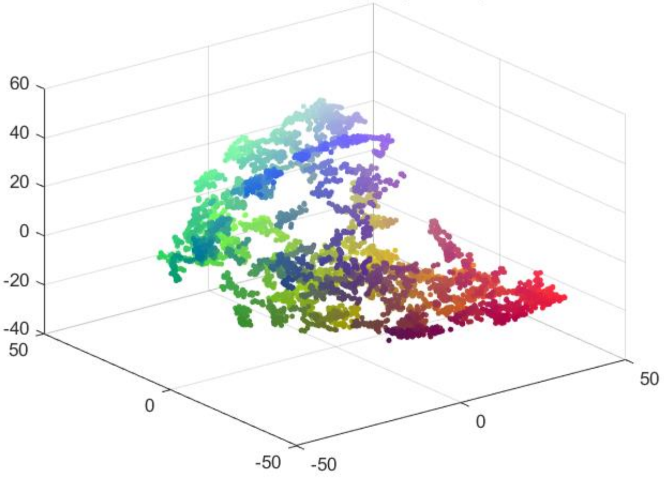
```
hexbin_size=80;
minimum_expression=5;
```

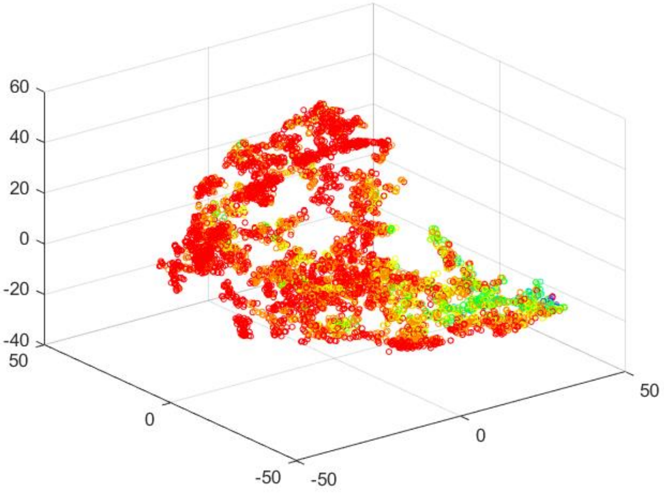
```
minimum_expression = 5
```

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

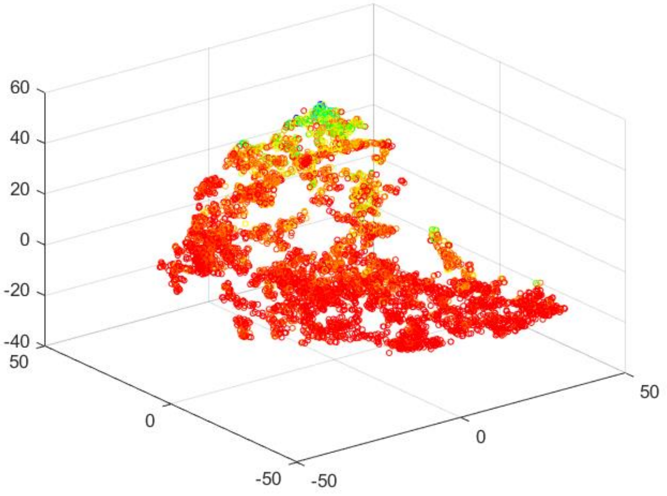
```
genes = 27×1 cell array
'AGT'
'APOE'
'AQP4'
'ARC'
'ASCL1'
'ATP1A2'
'BCAS1'
'BMPER'
'CCND1'
'EDNRB'
      ⋮
```


top two principle components

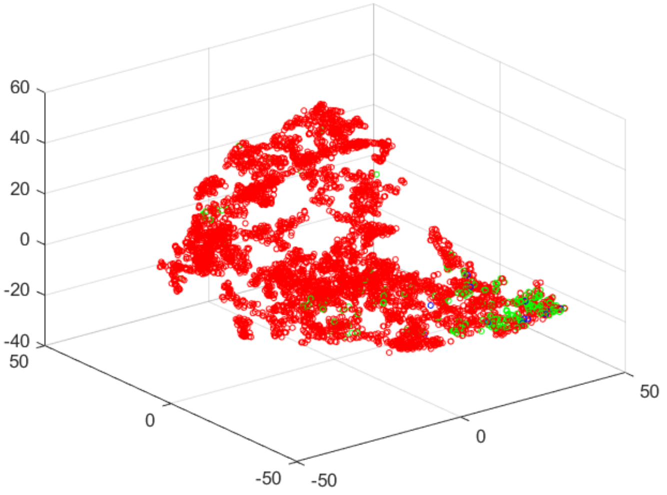

tSNE dim reduction to three
Color based on axis, according to spatial representation


AGT

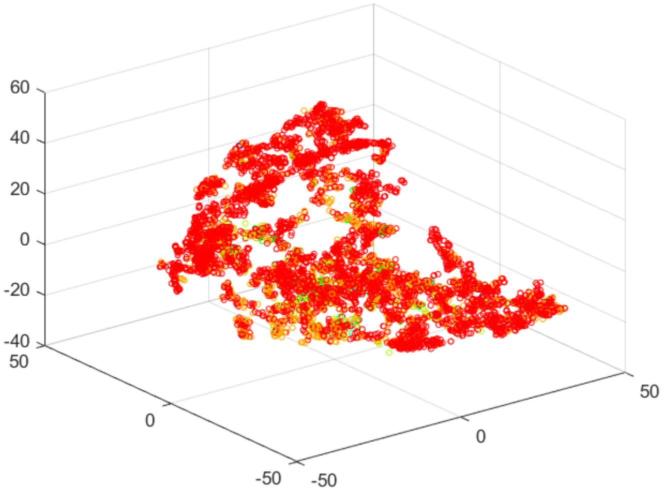
APOE



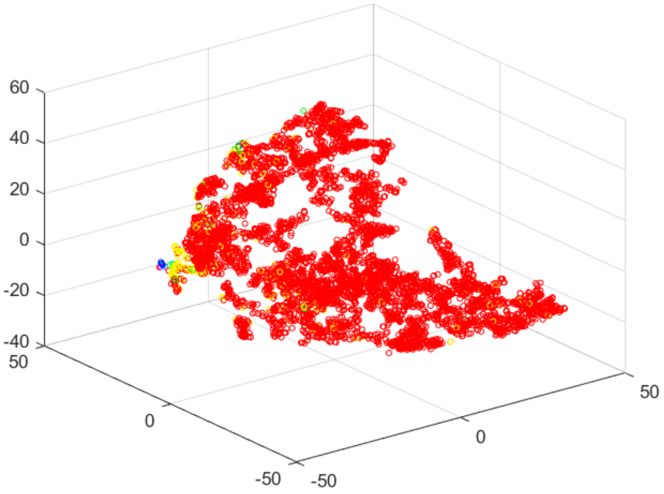
AQP4



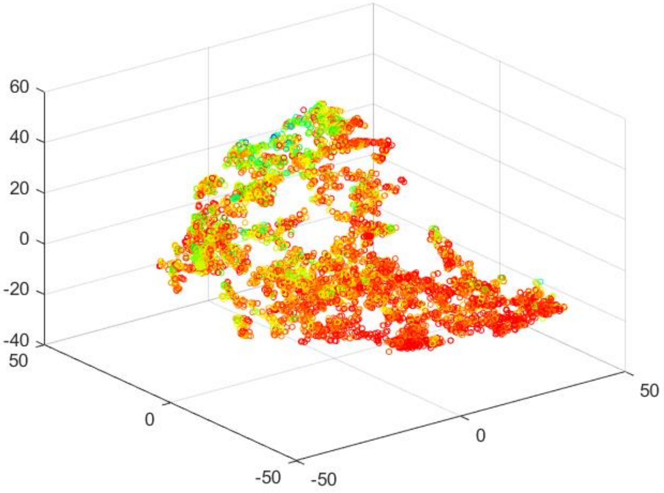
ARC



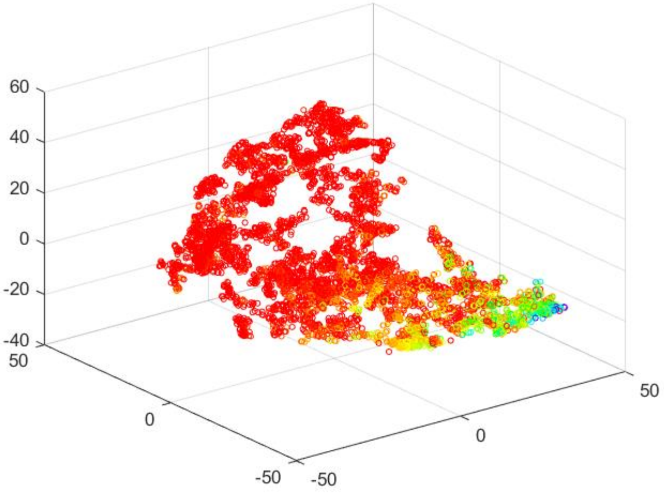
ASCL1



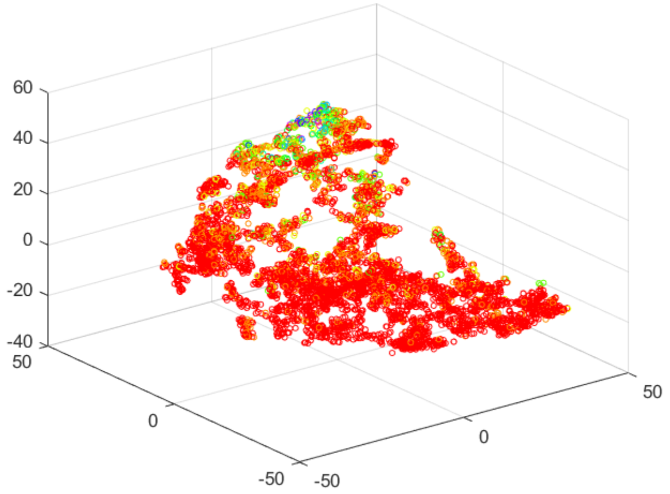
ATP1A2



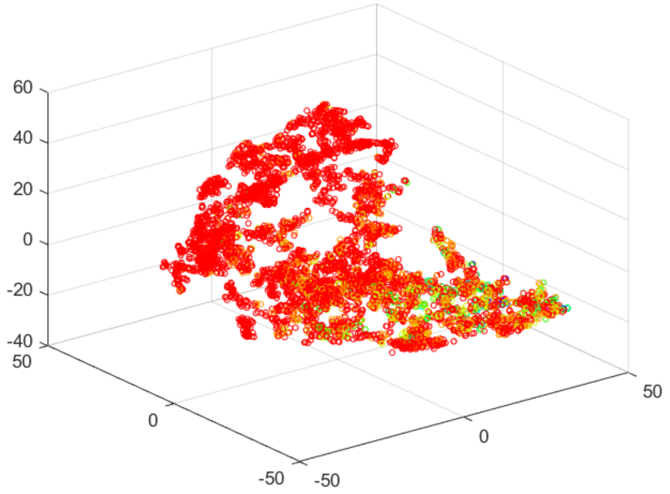
BCAS1



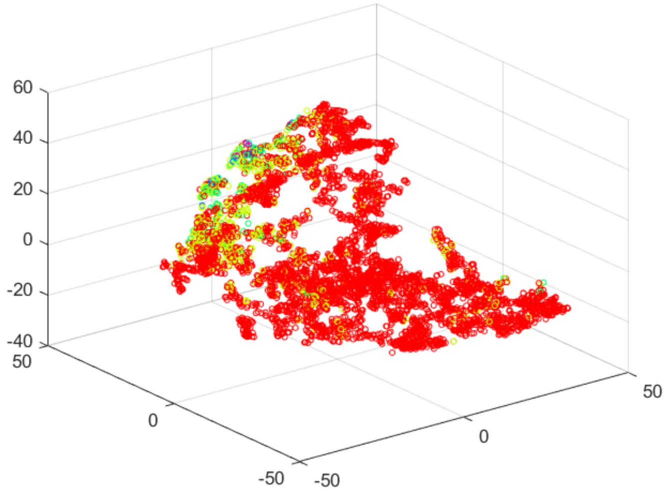
BMPER

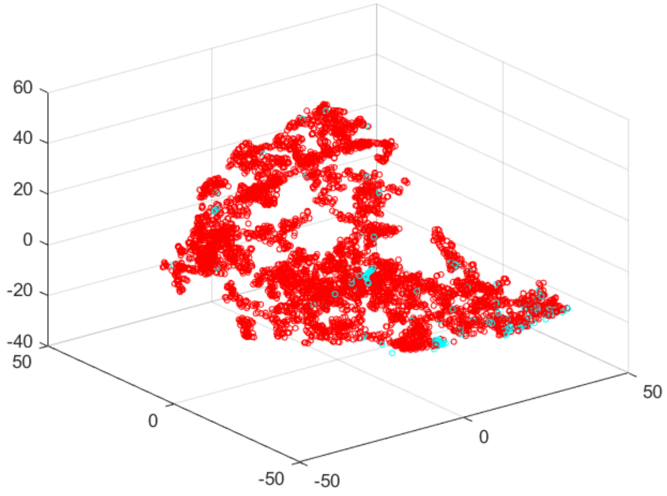


CCND1

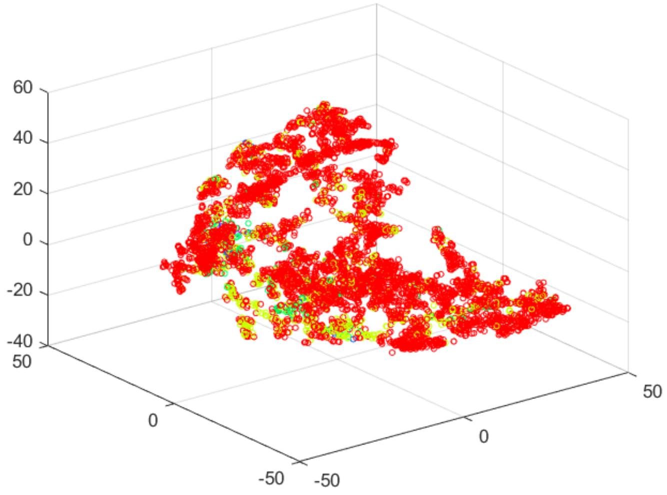


EDNRB

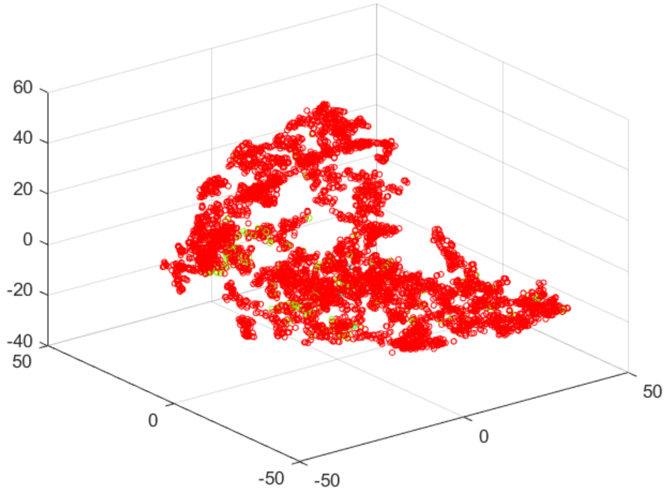




EGR1

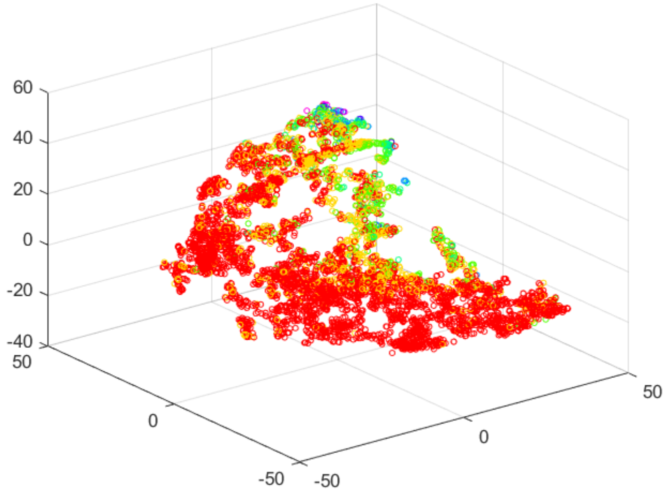


FYN

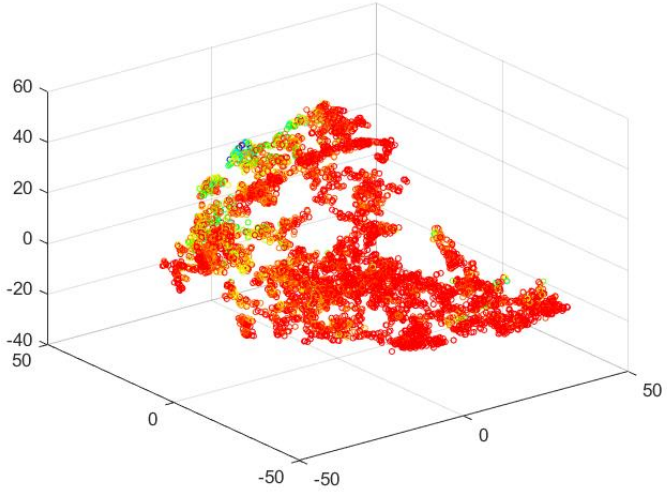


GFAP

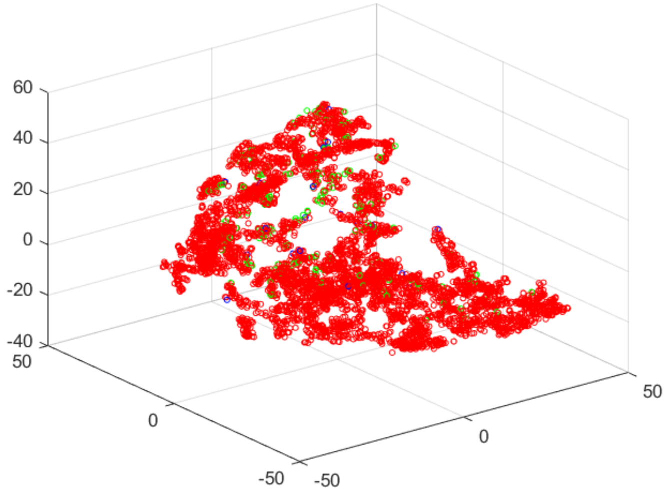
GPC3



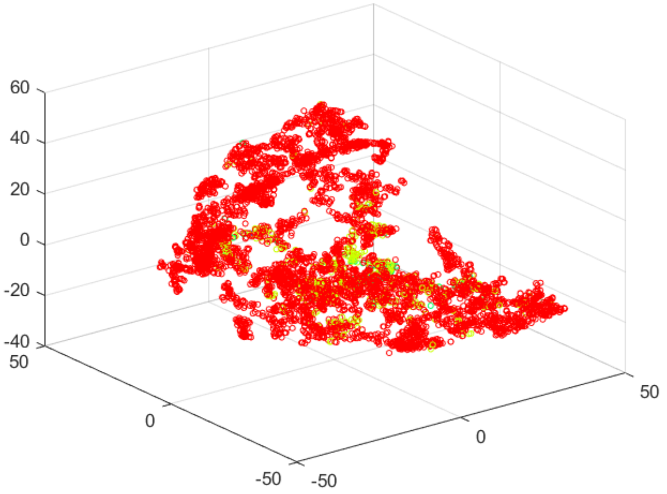
GPR17



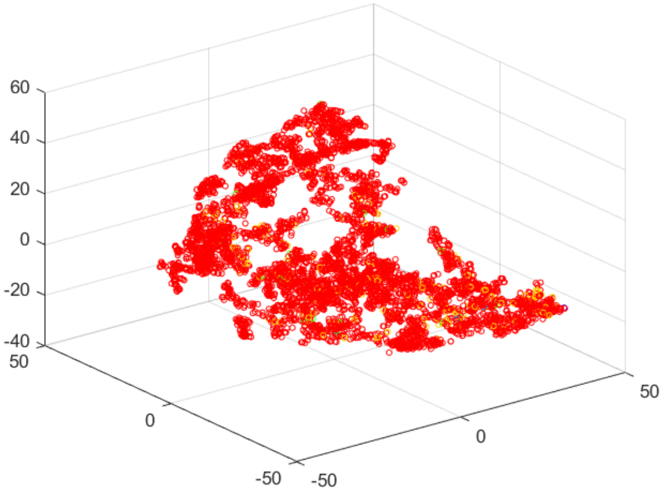
HOPX



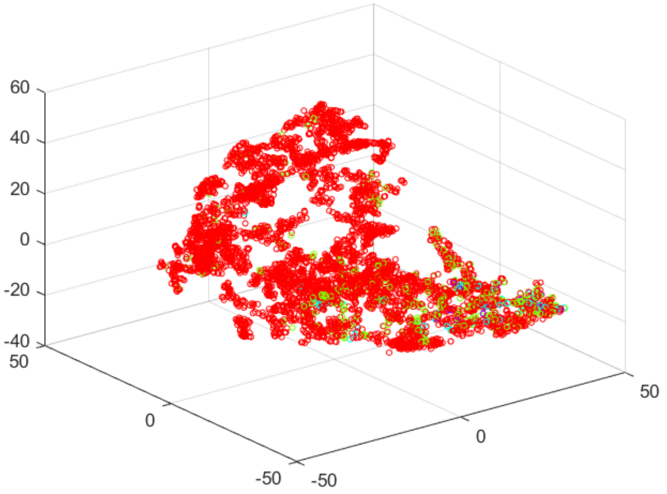
HSPB8

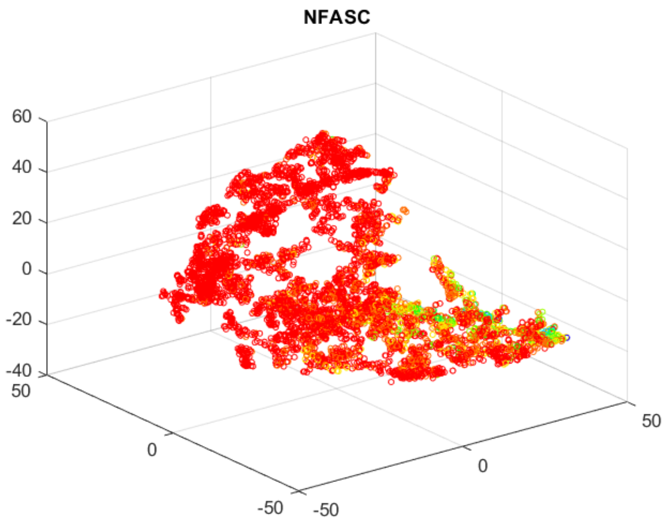


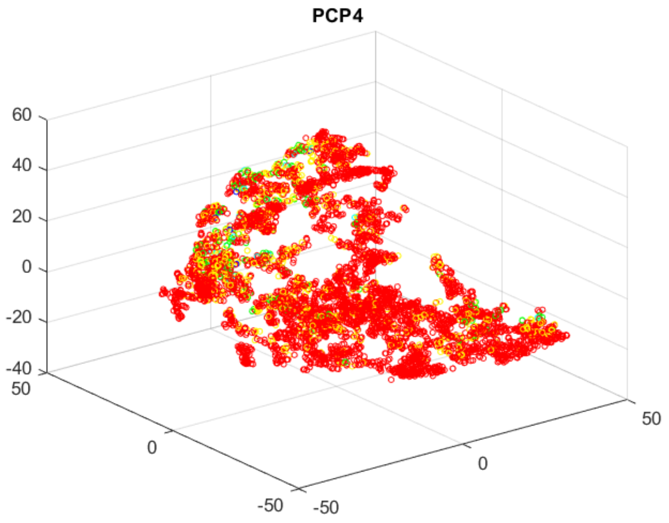
KLRC2

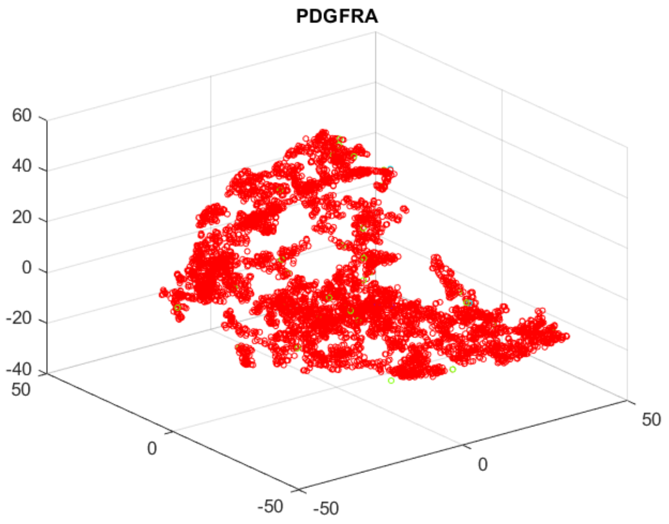


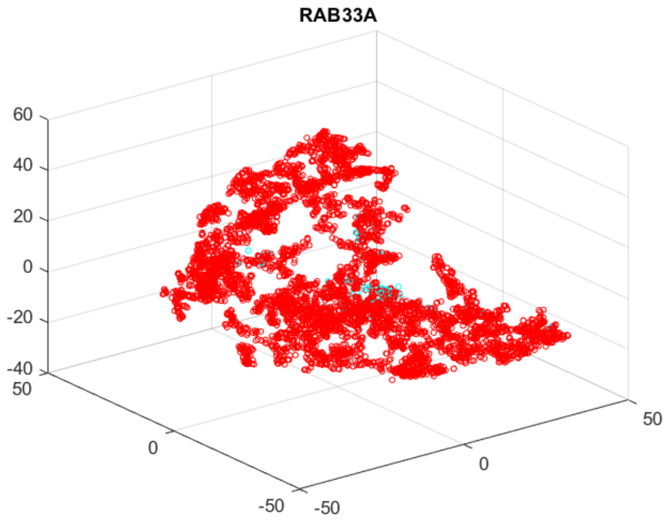
MEST

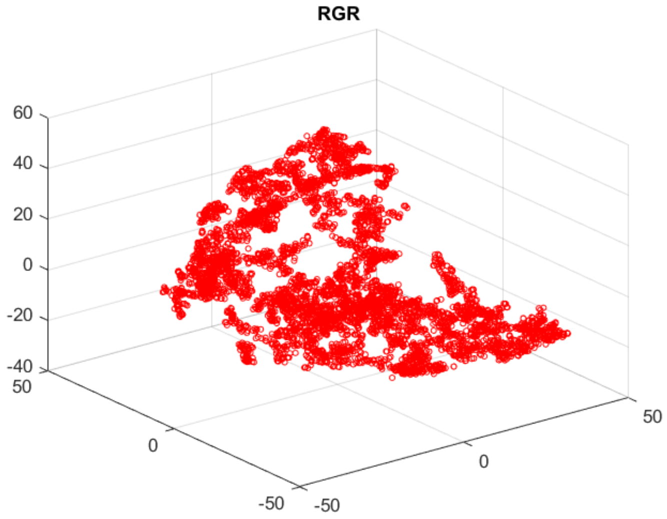


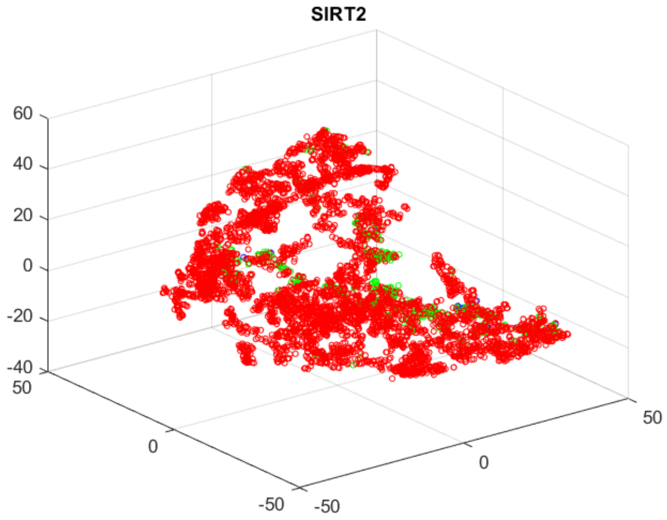

NFASC
60
40
20
0
-20
-40
50
0
-50
-50
0
50

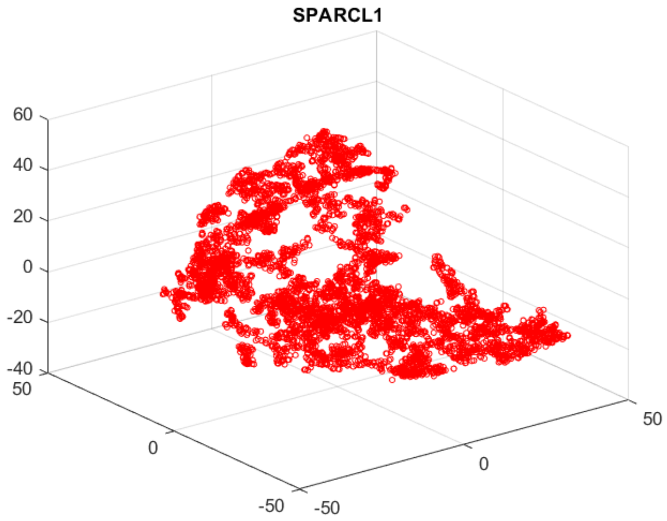

PCP4
60
40
20
0
-20
-40
50
0
-50
-50
0
50

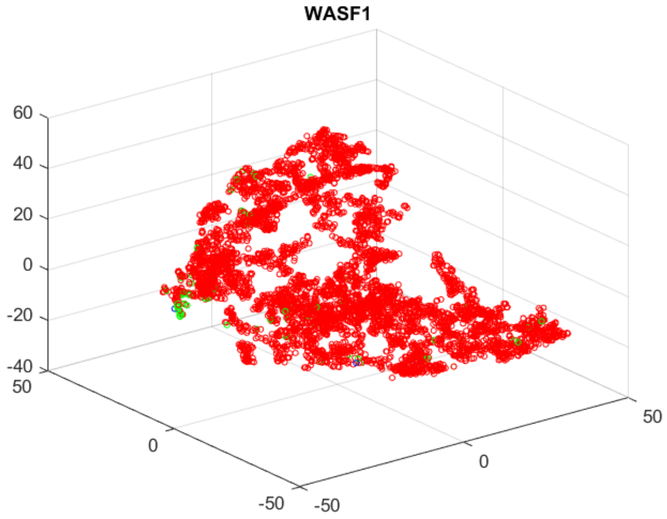

PDGFRA
60
40
20
0
-20
-40
50
0
-50
-50
0
50

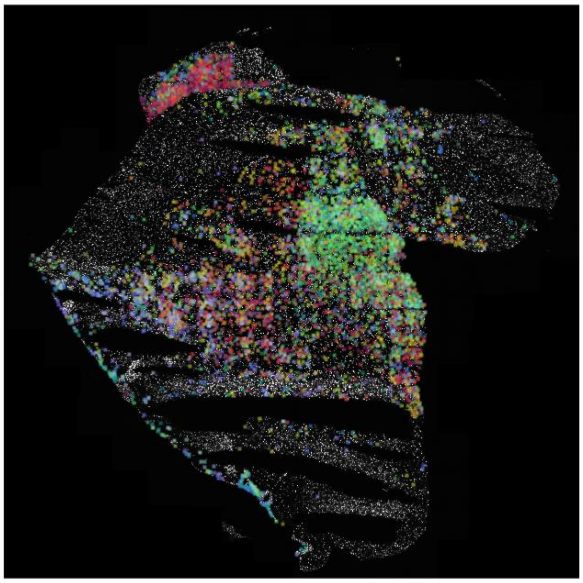

RAB33A
60
40
20
0
-20
-40
50
0
-50
-50
0
50


RGR
60
40
20
0
-20
-40
50
0
-50
-50
0
50


SIRT2
60
40
20
0
-20
-40
50
0
-50
-50
0
50


SPARCL1
60
40
20
0
-20
-40
50
0
-50
-50
0
50


WASF1
60
40
20
0
-20
-40
50
0
-50
-50
0
50



Hierarchical clustering data

Hierarchical custering performs a clustering based on a hierarchical strategy, by dividing data in binary categories consecutively. In this case, the number of categories desired needs to be specified by the user.

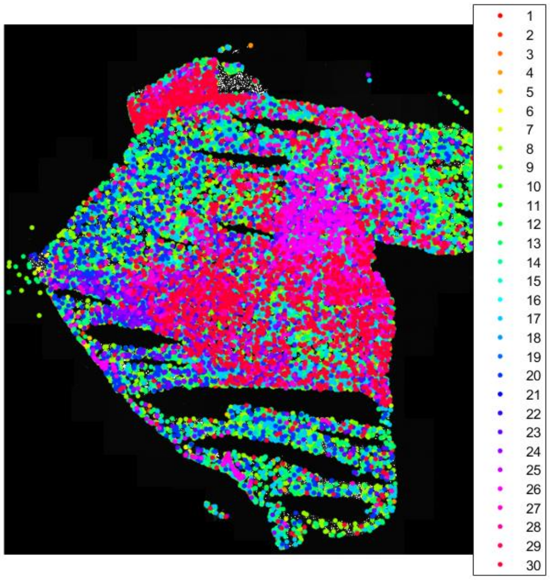
Create a subset of your genes (optional)

Selection of a subset of genes for the clustering, if desired.

```
%MATSEL.genename=EXPRESSIONMAT.genename(1:68);
%MATSEL.exp=EXPRESSIONMAT.exp(:,1:68);
%MATSEL.loc=EXPRESSIONMAT.loc;
%MATSEL.hexbinsize=EXPRESSIONMAT.hexbinsize;
```

Then we run hierarchical clustering.

```
CELLSHIERARCHICAL= ISS_hierarchical_clustering(EXPRESSIONMAT,SPOTS,30)
```

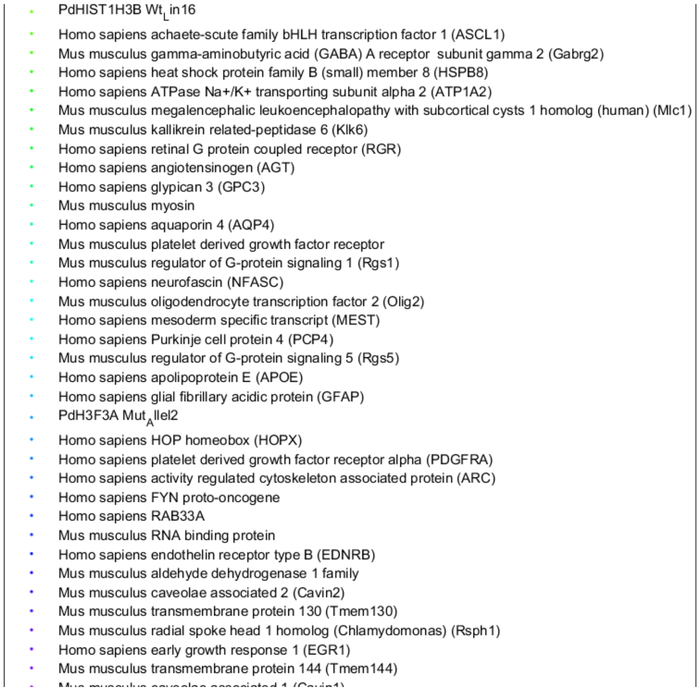


```
CELLSHIERARCHICAL = struct with fields:
    matrixcount: [23826×27 double]
       geneames: {27×1 cell}
           name: [1×23826 double]
            pos: [23826×2 double]
          image: 'G:\DIPG pciseq\Sample2\Base_2_aligned-1.tif'
          scale: 1
```

This part plots the overall gene expression on top of the cell-type distribution plot.

In case we want to superposethis segmentation with Spots, add this.

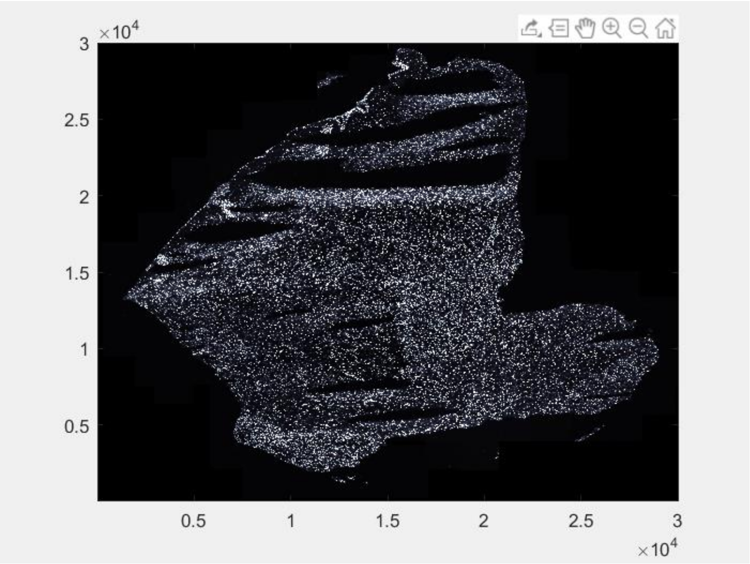
```
hold on
gscatter(SPOTS.pos(:,1),SPOTS.pos(:,2),SPOTS.name);
hold on
 %RGB=rgbscale(CELLS.Probabilities(:,1:3));
 gscatter(CELLS.pos(:,1),CELLS.pos(:,2),CELLS.name(1:end,:))
 scatter(CELLS.pos(:,1),CELLS.pos(:,2),10,RGB(1:end-1,:),'filled');
```



PCISEQ (Probabilistic cell typing)

PCIseq performs cell type assigment using single cell RNAseq and assuming that the sample contains the same cell types than the scRNAseq dataset. First we need to define a region.

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



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

Then, we load the scRNAseq data already clustered on our gSet object. This will be used as an input in pciSeq

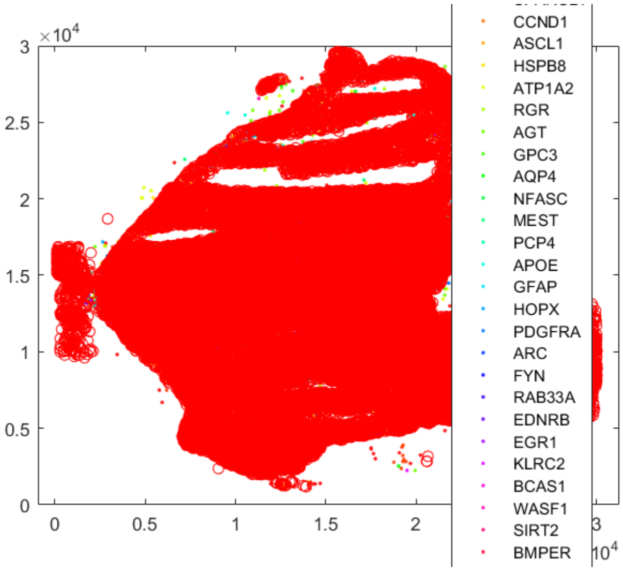
Remember to load o to path

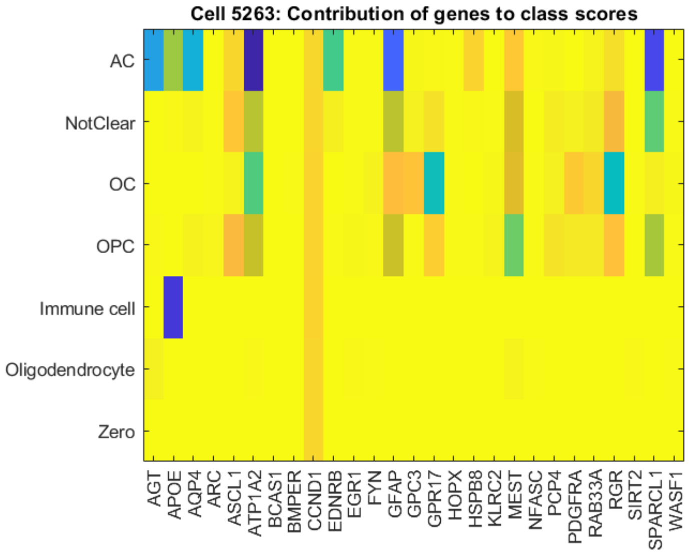
This is gSet for DIPG

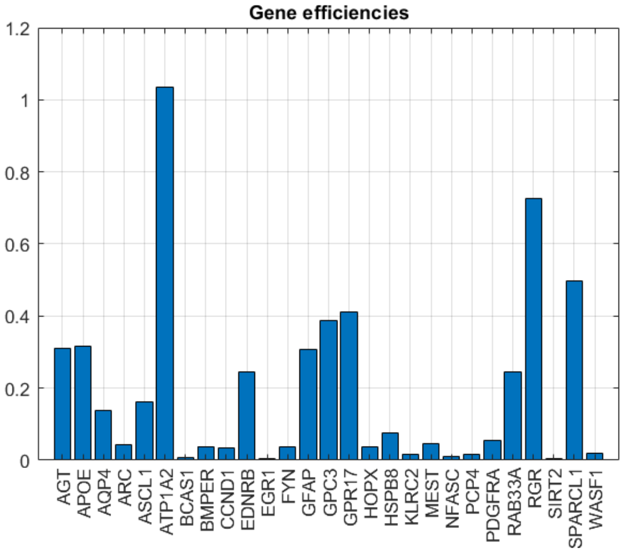
```
load E:\pciSeq\Data\AllSections\gSetCA1all.mat;
gSet.GeneExp=readmatrix('G:\DIPG pciseq\expressionmatrix_selected.csv');
gSet.GeneExp=gSet.GeneExp(:,3:end);
nam=readcell('G:\DIPG pciseq\expressionmatrix_selected.csv');
gSet.GeneName=nam(2:end,2);
namclust=readcell('G:\DIPG pciseq\metadata_selected.csv');
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;
```

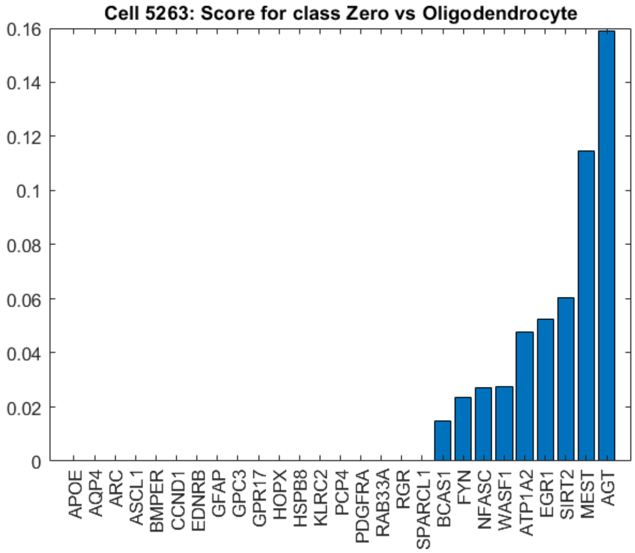
Running PCIseq itself

```
  CELLS = ISS_pciseq(SPOTS,CellMap,gSet)
```


×10^4
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
3
×10^4
CCND1
ASCL1
HSPB8
ATP1A2
RGR
AGT
GPC3
AQP4
NFASC
MEST
PCP4
APOE
GFAP
HOPX
PDGFRA
ARC
FYN
RAB33A
EDNRB
EGR1
KLRC2
BCAS1
WASF1
SIRT2
BMPER


Cell 5263: Contribution of genes to class scores
AC
NotClear
OC
OPC
Immune cell
Oligodendrocyte
Zero
AGT
APOE
AQP4
ARC
ASCL1
ATP1A2
BCAS1
BMPER
CCND1
EDNRB
EGR1
FYN
GFAP
GPC3
GPR17
HOPX
HSPB8
KLRC2
MEST
NFASC
PCP4
PDGFRA
RAB33A
RGR
SIRT2
SPARCL1
WASF1

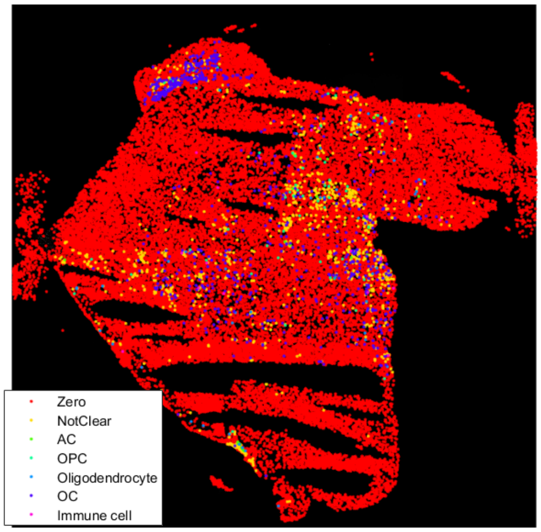

Gene efficiencies
1.2
1
0.8
0.6
0.4
0.2
0
AGT
APOE
AQP4
ARC
ASCL1
ATP1A2
BCAS1
BMPER
CCND1
EDNRB
EGR1
FYN
GFAP
GPC3
GPR17
HOPX
HSPB8
KLRC2
MEST
NFASC
PCP4
PDGFRA
RAB33A
RGR
SIRT2
SPARCL1
WASF1



```
CELLS = struct with fields:
      matrixcount: [44494×27 double]
         geneames: {27×1 cell}
             name: {44494×1 cell}
              pos: [44494×2 double]
       ClassNames: {7×1 cell}
    Probabilities: [44495×7 double]
            image: 'G:\DIPG pciseq\Sample2\Base_2_aligned-1.tif'
            scale: 1
```

Plotting the output of PCIseq

```
figure
imshow(imread(SPOTS.image))
hold on
%RGB=rgbscale(CELLS.Probabilities(:,1:3));
gscatter(CELLS.pos(:,1),CELLS.pos(:,2),CELLS.name(1:end,:))
```

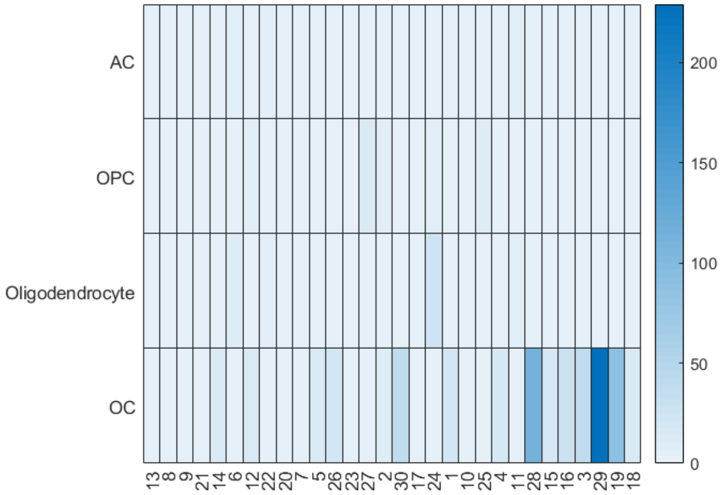


```
%scatter(CELLS.pos(:,1),CELLS.pos(:,2),10,RGB(1:end-1,:),'filled');
```

Comparison between PCIseq and hierarchical Clustering

Selection based on CELS

```
SELECTED.clusters=cellstr(num2str(CELLSHIERARCHICAL.name'));
SELECTED.pos=CELLS.pos(CELS,:);
SELECTED.name=CELLS.name(CELS,:);

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

figure
XC=unique(SELECTED.clusters);
YC=unique(SELECTED.name);
h = heatmap(labels(:,2),labels(3:6,1),celltypecor(3:6,:))
```



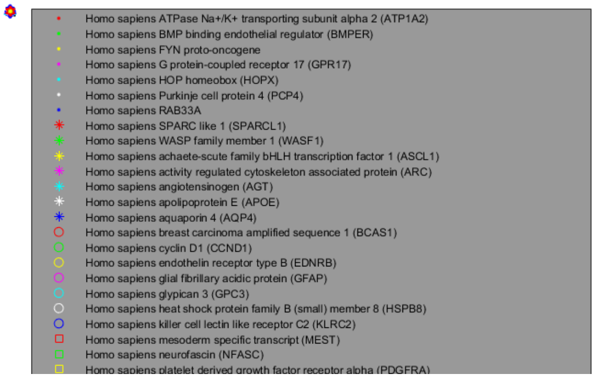
```
h = 
  HeatmapChart with properties:

        XData: {30×1 cell}
        YData: {4×1 cell}
    ColorData: [4×30 double]

  Show all properties
```

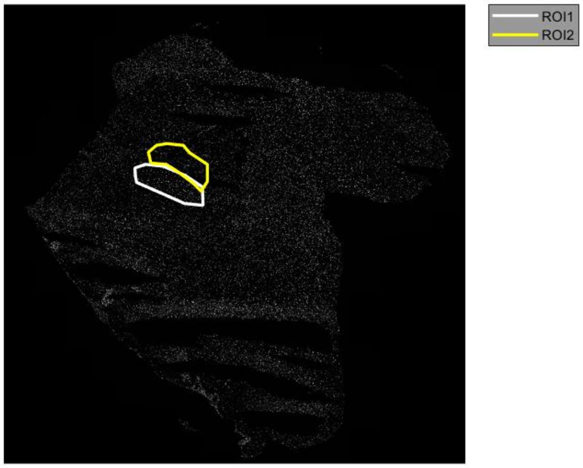
ROI selection

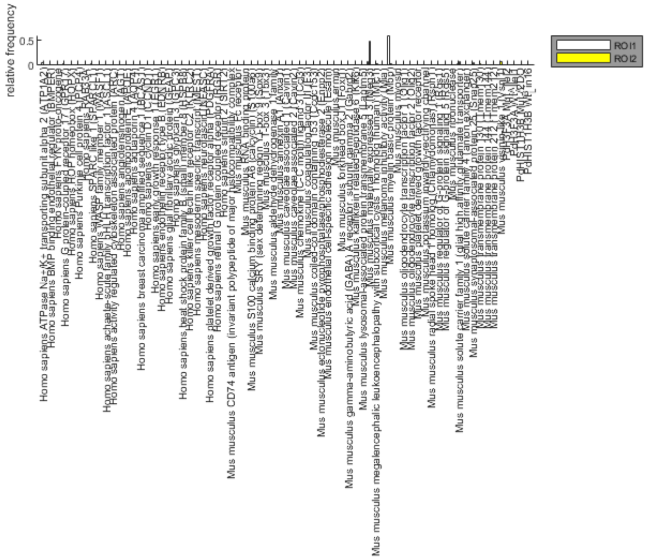
We select diferent ROI and we save this subobjects to INROI

```
 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 'Mus musculus myelin basic protein (Mbp)',100)
 %OR DAPI
 roi_number=2;
 [INROI,ROI_borders] = ISS_ROI_draw_onImage(SPOTS,output_directory,roi_number,'Mus musculus myelin basic protein (Mbp)',100);
```

```
loading image..
Warning: Directory already exists.
```


Homo sapiens ATPase Na+/K+ transporting subunit alpha 2 (ATP1A2)
Homo sapiens BMP binding endothelial regulator (BMPER)
Homo sapiens FYN proto-oncogene
Homo sapiens G protein-coupled receptor 17 (GPR17)
Homo sapiens HOP homeobox (HOPX)
Homo sapiens Purkinje cell protein 4 (PCP4)
Homo sapiens RAB33A
Homo sapiens SPARC like 1 (SPARCL1)
Homo sapiens WASP family member 1 (WASF1)
Homo sapiens achaete-scute family bHLH transcription factor 1 (ASCL1)
Homo sapiens activity regulated cytoskeleton associated protein (ARC)
Homo sapiens angiotensinogen (AGT)
Homo sapiens apolipoprotein E (APOE)
Homo sapiens aquaporin 4 (AQP4)
Homo sapiens breast carcinoma amplified sequence 1 (BCAS1)
Homo sapiens cyclin D1 (CCND1)
Homo sapiens endothelin receptor type B (EDNRB)
Homo sapiens glial fibrillary acidic protein (GFAP)
Homo sapiens glypican 3 (GPC3)
Homo sapiens heat shock protein family B (small) member 8 (HSPB8)
Homo sapiens killer cell lectin like receptor C2 (KLRC2)
Homo sapiens mesoderm specific transcript (MEST)
Homo sapiens neurofascin (NFASC)
Homo sapiens platelet derived growth factor receptor alpha (PDGFRA)

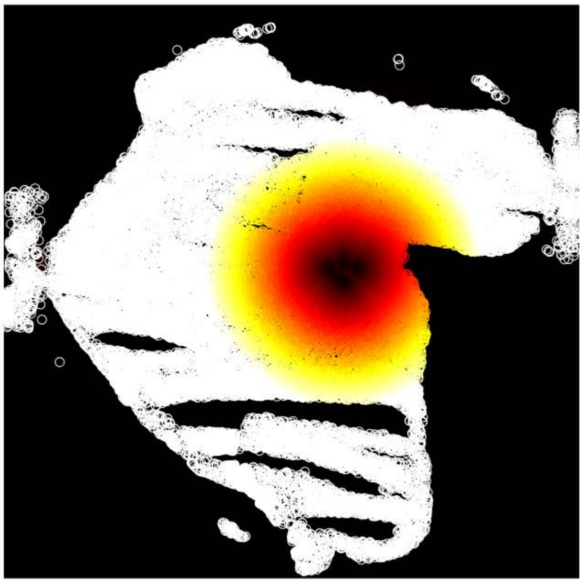

ROI1
ROI2

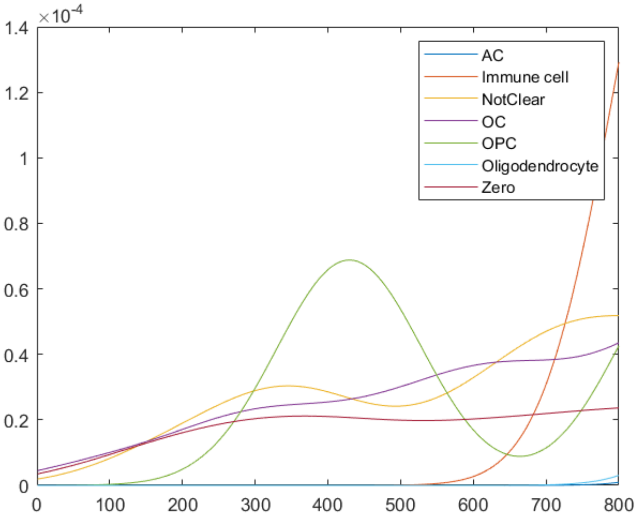


GRADIENT ANALYSIS

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

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



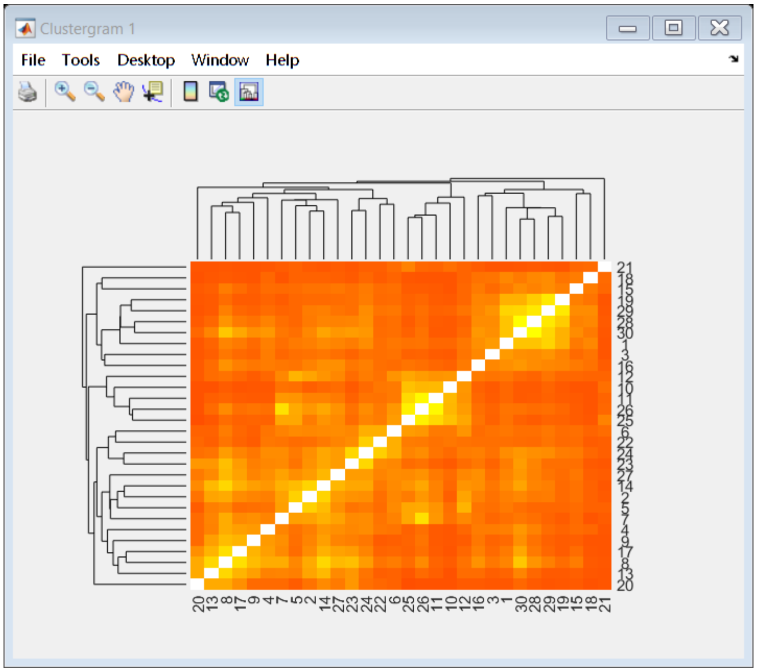


Seach for the closest class in terms of expression

It calculates the density for all the genes/cells and it computes the correlation between them.

NOTE: It can be a slow process. The bandwidth specified is a quite key parameter

```
bandwidth=100;
[MEANDIST] =ISS_densities_correlation(CELLSHIERARCHICAL,bandwidth)
```

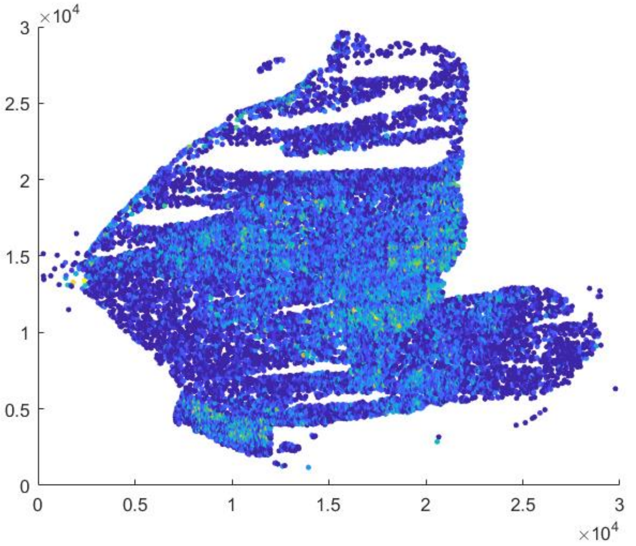


```
MEANDIST = 30×30
    1.0000    0.1236    0.1962    0.1240    0.1061    0.0997    0.0753 ...
    0.1236    1.0000    0.0853    0.1173    0.3643    0.1464    0.2250
    0.1962    0.0853    1.0000    0.1118    0.0727    0.0830    0.0507
    0.1240    0.1173    0.1118    1.0000    0.0790    0.0880    0.1365
    0.1061    0.3643    0.0727    0.0790    1.0000    0.1486    0.2166
    0.0997    0.1464    0.0830    0.0880    0.1486    1.0000    0.1328
    0.0753    0.2250    0.0507    0.1365    0.2166    0.1328    1.0000
    0.1345    0.2426    0.0961    0.2554    0.1415    0.1096    0.2569
    0.0616    0.1353    0.0417    0.2003    0.0731    0.0300    0.1701
    0.0511    0.0236    0.0134    0.0017    0.0721    0.1055    0.0806
      ⋮
```

HETEROGENEITY MAP

Then we plot heterogeneity within the tissue. The more yellow, the more homogeneus a tissue is.

```
ISS_heterogeneity_map(EXPRESSIONMAT,SPOTS)
```


×10^4
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
×10^4