# Giotto: a toolbox for integrative analysis and visualization of spatial expression data

Guo-Cheng Yuan Lab

2020.06.11

# Introduction to Yuan Lab

Lab Website: http://ds.dfci.harvard.edu/~gcyuan

Dana-Farber, Harvard Medical School

Ruben Dries

Qian Zhu

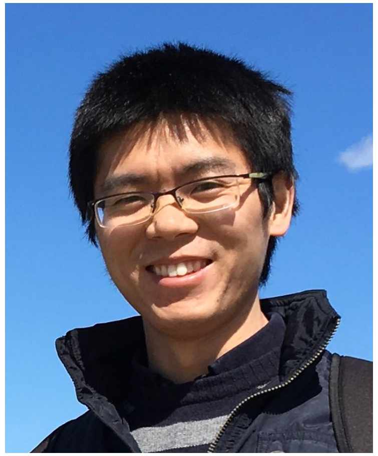
Rui Dong

Huipeng Li

Arpan Sarkar

Shiwei Zheng

Shengbao Suo

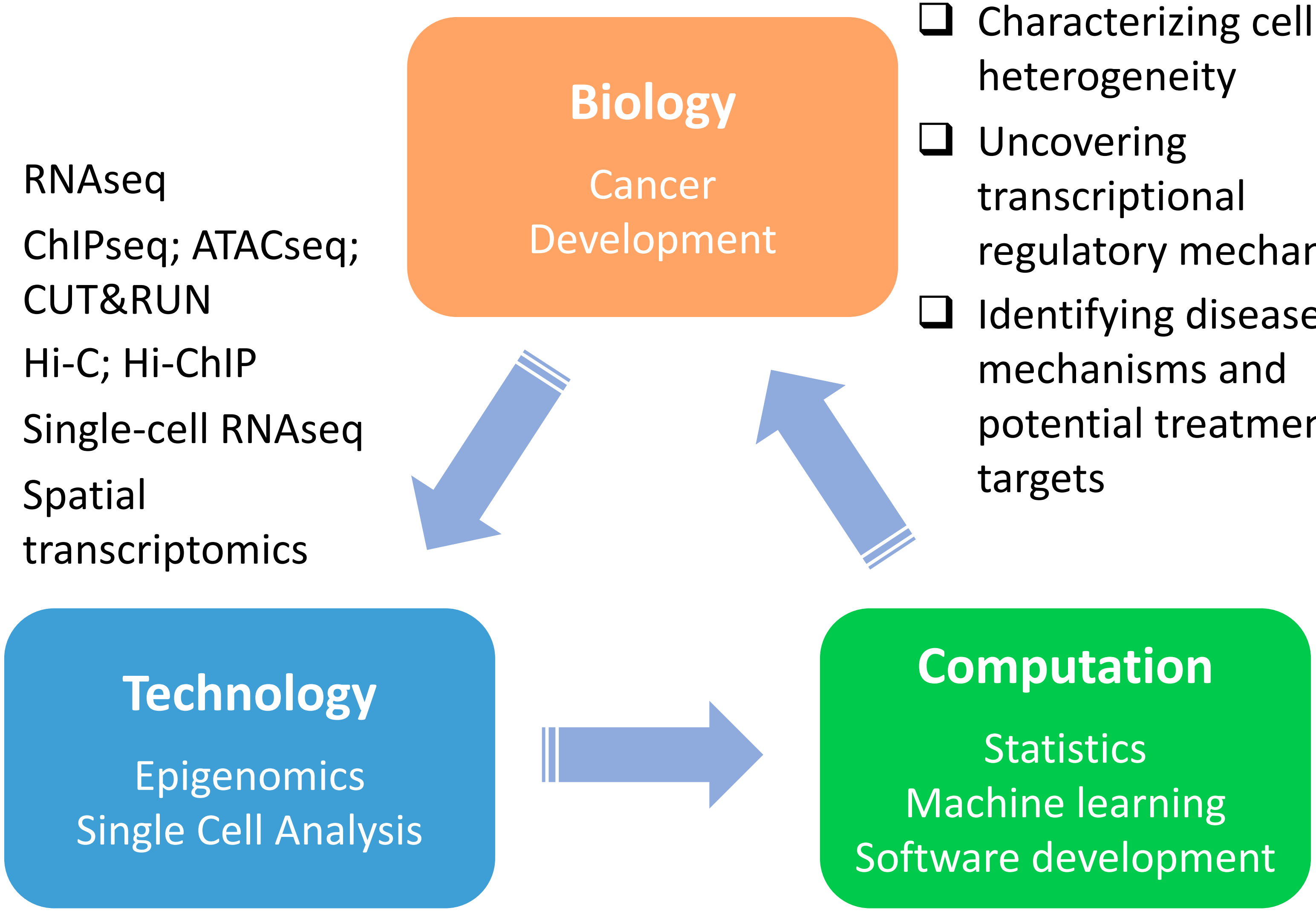


Yan Kai

Fulong Yu

# Research interests

- ❑ RNAseq
- ❑ ChIPseq; ATACseq; CUT&RUN
- ❑ Hi-C; Hi-ChIP
- ❑ Single-cell RNAseq
- ❑ Spatial transcriptomics



**Biology**

Cancer Development

**Technology**

Epigenomics
Single Cell Analysis

**Computation**

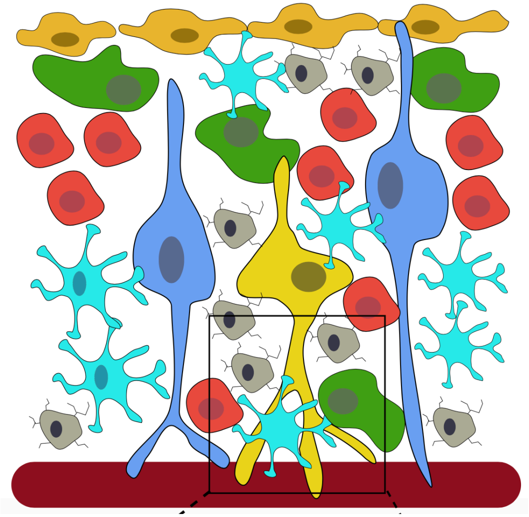
Statistics
Machine learning
Software development

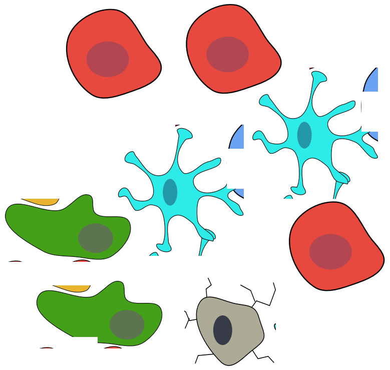
- ❑ Characterizing cellular heterogeneity
- ❑ Uncovering transcriptional regulatory mechanisms.
- ❑ Identifying disease mechanisms and potential treatment targets

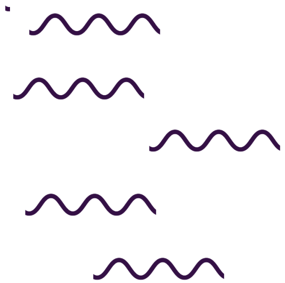
- ❑ Haystack; diHMM; CRISPResso; CUT&RUNTools
- ❑ SCUBA; GiniClust; DWLS; Giotto

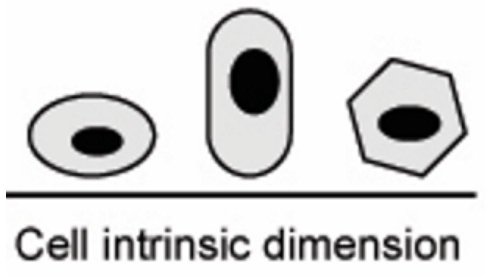
# Characterizing cellular heterogeneity

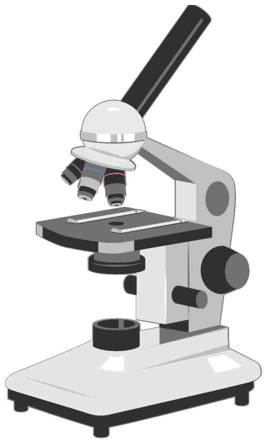
scRNA-seq

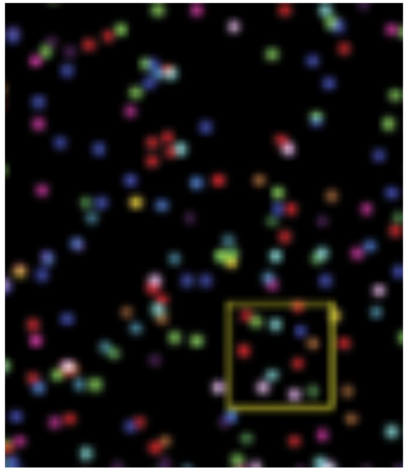


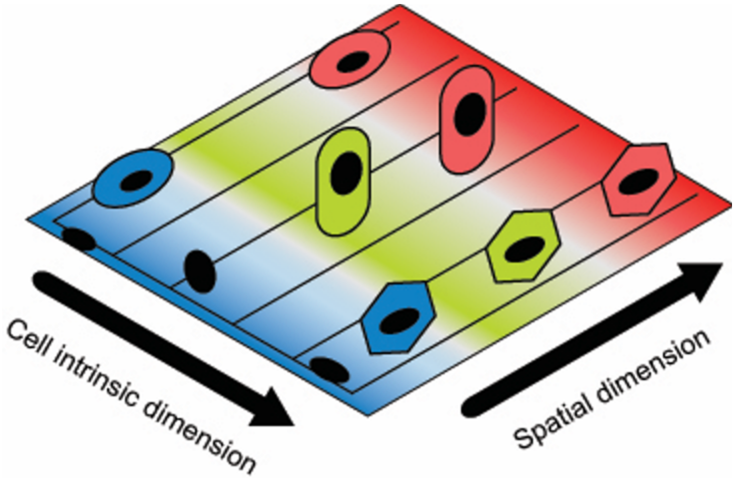








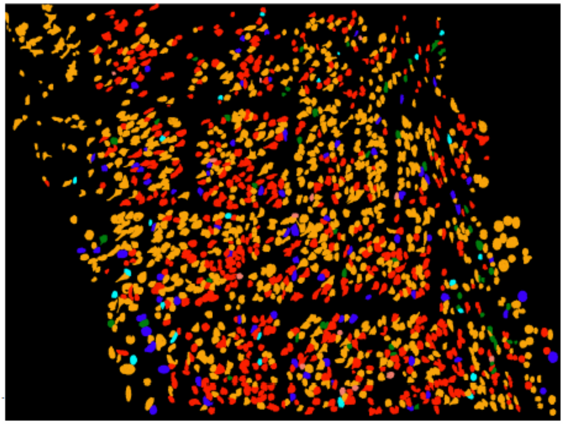




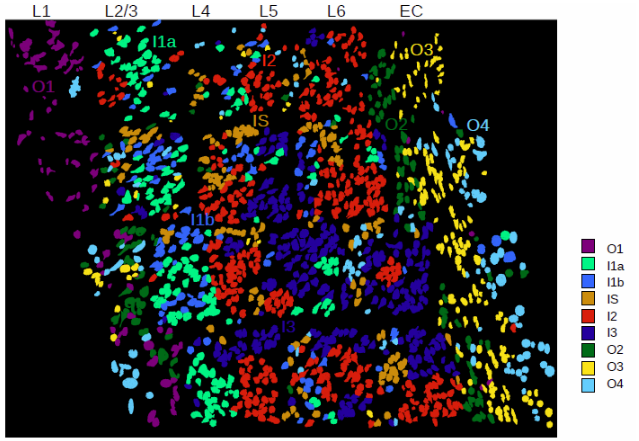
spatial transcriptomics

# Previous Work

Cell Type

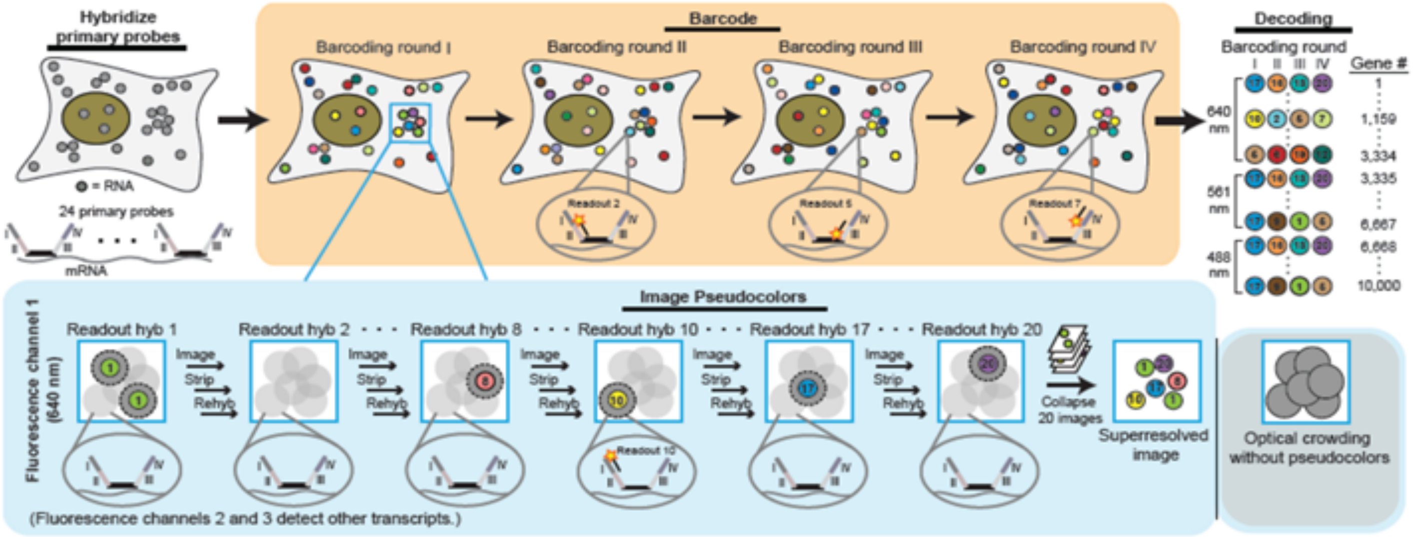


Spatial Domain



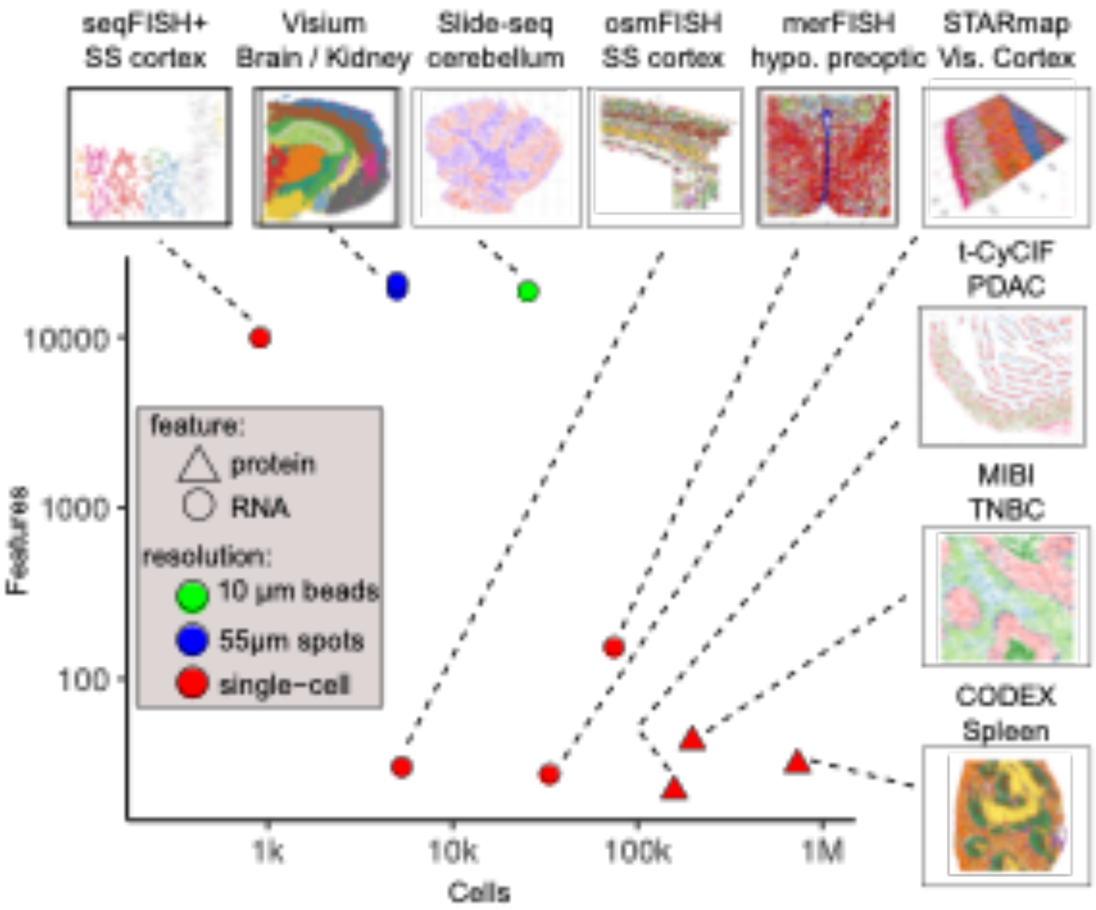
Zhu et al. Nature Biotech, 2018

In Collaboration with Long Cai’s Lab



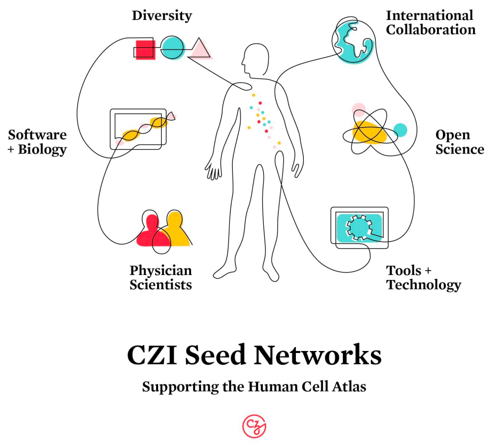
Eng et al. Nature, 2019

# Why we develop Giotto



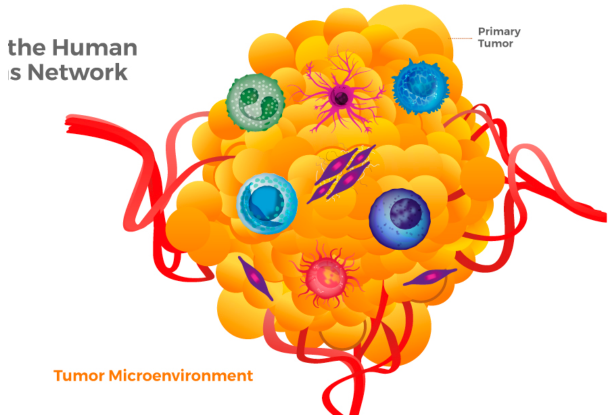
HUMAN CELL ATLAS

HuBMAP
The Human BioMolecular Atlas Program

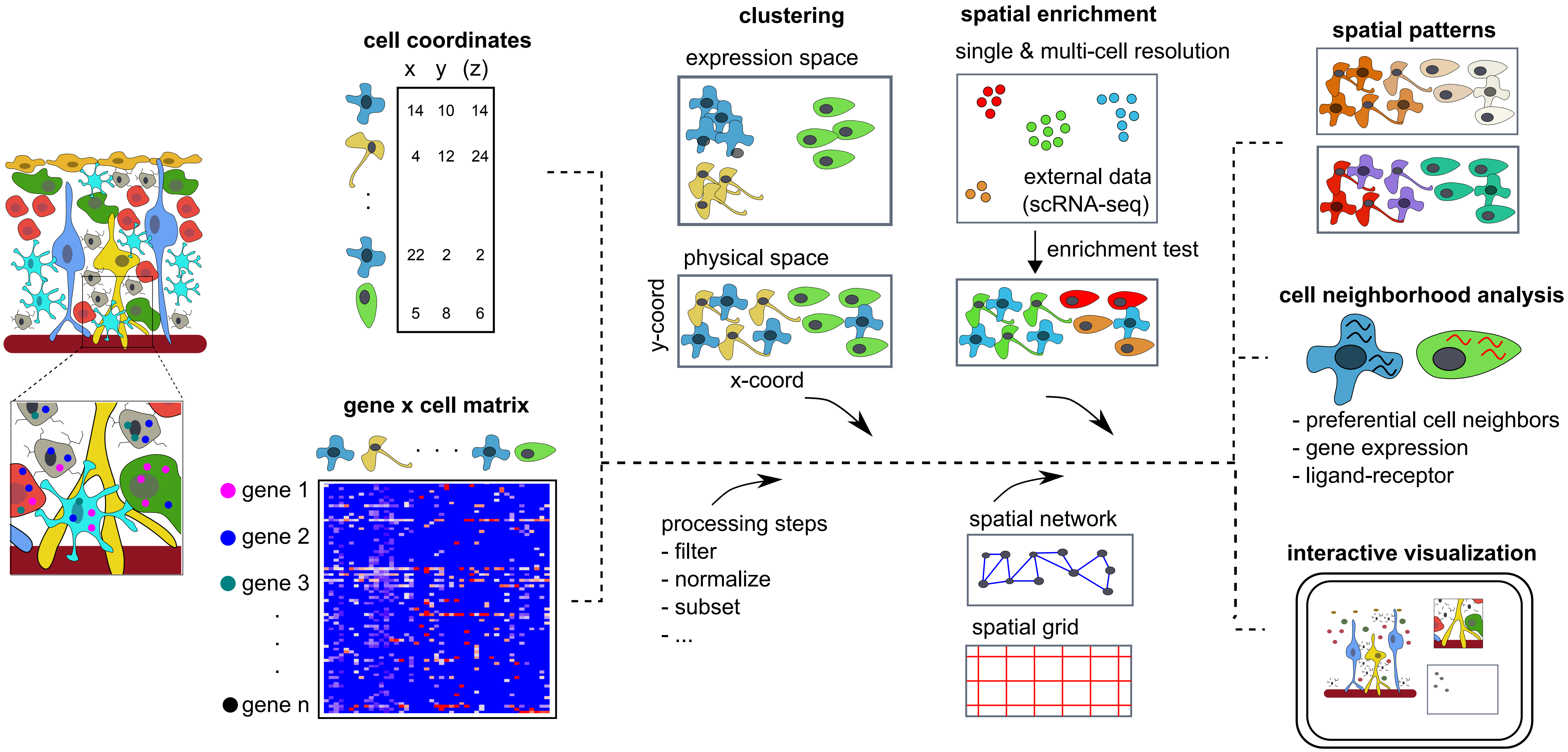


CZI Seed Networks

Supporting the Human Cell Atlas



# Giotto in a snapshot



**cell coordinates**

| x | y | (z) |
|---|---|---|
| 14 | 10 | 14 |
| 4 | 12 | 24 |
| 22 | 2 | 2 |
| 5 | 8 | 6 |

**gene x cell matrix**

- gene 1
- gene 2
- gene 3
- gene n

**clustering**

expression space

physical space

y-coord

x-coord

processing steps
- filter
- normalize
- subset
- ...

**spatial enrichment**

single & multi-cell resolution

external data (scRNA-seq)

enrichment test

spatial network

spatial grid

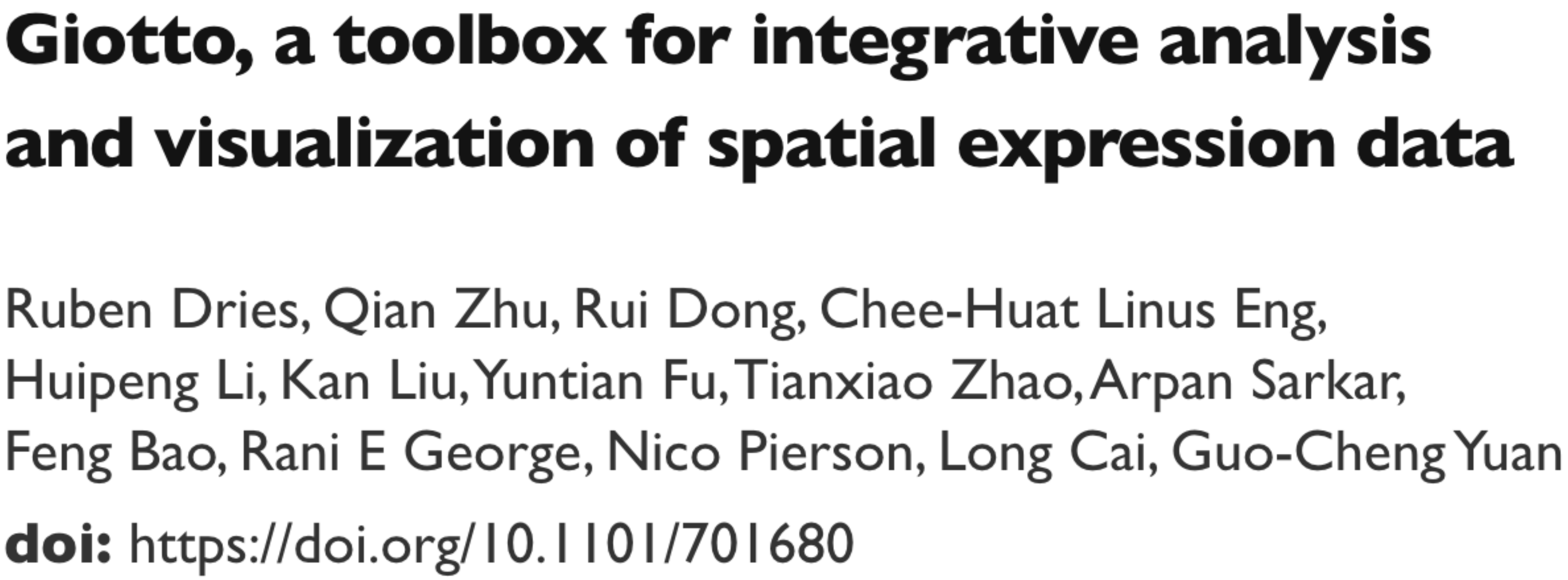
**spatial patterns**

**cell neighborhood analysis**

- preferential cell neighbors
- gene expression
- ligand-receptor

**interactive visualization**

# Key publication and websites



**Giotto, a toolbox for integrative analysis and visualization of spatial expression data**

Ruben Dries, Qian Zhu, Rui Dong, Chee-Huat Linus Eng,
Huipeng Li, Kan Liu, Yuntian Fu, Tianxiao Zhao, Arpan Sarkar,
Feng Bao, Rani E George, Nico Pierson, Long Cai, Guo-Cheng Yuan

**doi:** https://doi.org/10.1101/701680

**Main Website:** [www.spatialgiotto.com](www.spatialgiotto.com)

**Development Website:**

**Giotto Analyzer:** [https://rubd.github.io/Giotto_site/](https://rubd.github.io/Giotto_site/)

**Giotto Viewer:** [http://spatialgiotto.rc.fas.harvard.edu/giotto-viewer/](http://spatialgiotto.rc.fas.harvard.edu/giotto-viewer/)

# Goal of Today

Overview of the Giotto pipeline

Describe the major functionalities in Giotto

Explain how Giotto can be adapted to address platform-specific needs

Demonstrate the utility of Giotto using various datasets from a number of technologies

# Agenda and Logistics

## Specific Topics

Overview of the Giotto toolbox

Cell-type identification and data visualization

Analysis of data with non-single-cell spatial resolution

Spatial patterns and cell neighborhood network analyses

Interactive data visualization

## Logistics

Questions can be submitted via chat throughout this presentation.

Additional time for live Q&A session
- please use the ‘raise hand’ feature

This presentation will be recorded and posted online.

All slides will be uploaded and posted online.

# Giotto 101

## Giotto 101: Start with Giotto

**Ruben Dries**

*@RnDries*

# Giotto 101

1. How to install Giotto?

2. How does a typical Giotto analysis looks like?

# How to install Giotto

**Giotto Analyzer**

Data analysis and visualization

R package

**Giotto Viewer**

Interactive visualization

local application in your browser (e.g. chrome)

# How to install Giotto Analyzer

**Giotto Analyzer**

Data analysis and visualization

R package

Currently:

```
library(devtools)
library(remotes)
remotes::install_github(“RubD/Giotto”)
```

Future (CRAN):

```
install.packages(“Giotto”)
```

# How to install python modules

**Giotto Analyzer**

Data analysis and visualization

R package

Python dependencies

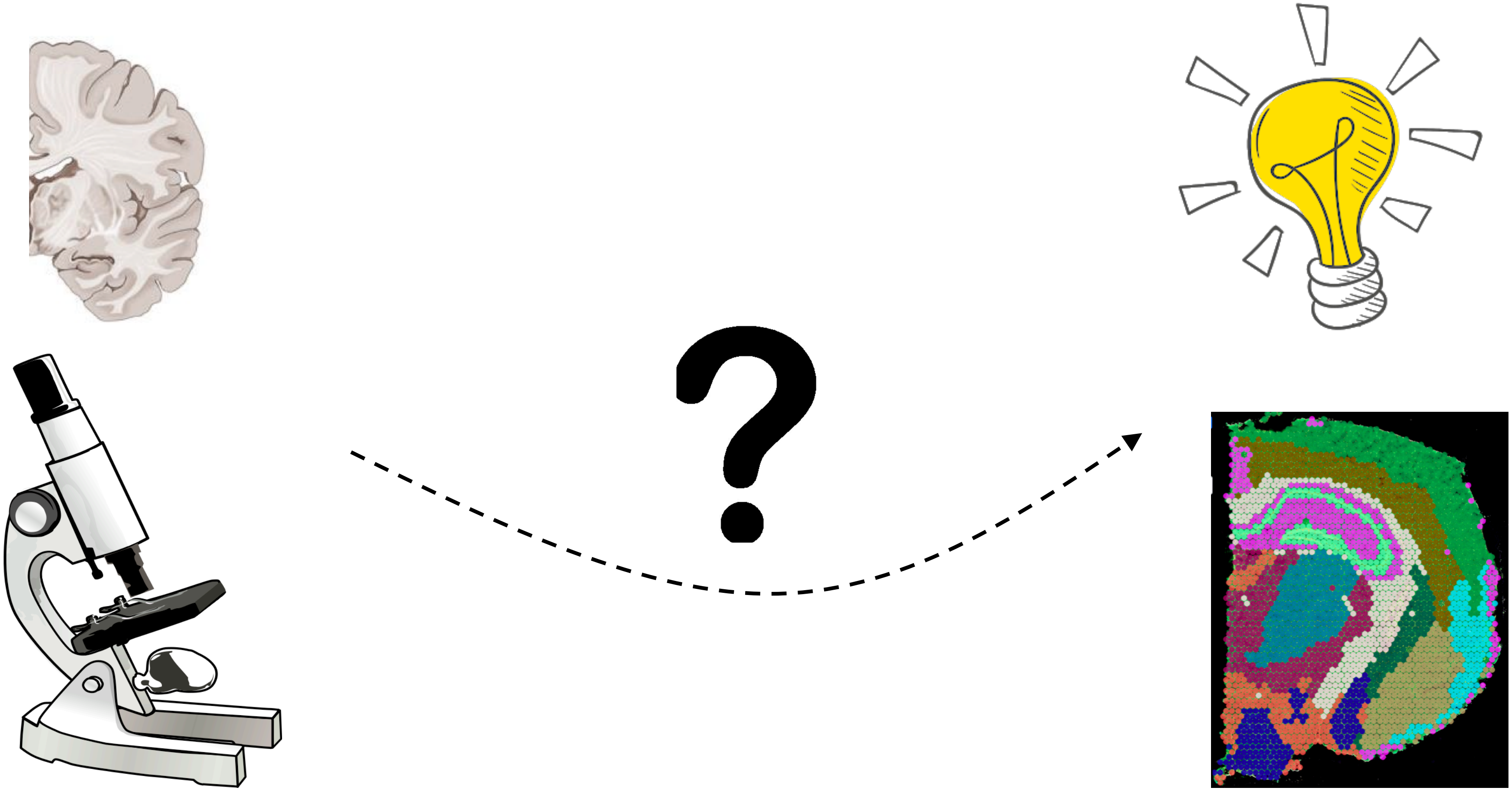
**Automatic installation:**
First time you run Giotto it will create a Giotto environment with all necessary python modules

**Alternative manual installation:**
You can install the necessary python modules yourself however you want (e.g. conda) and provide the python path

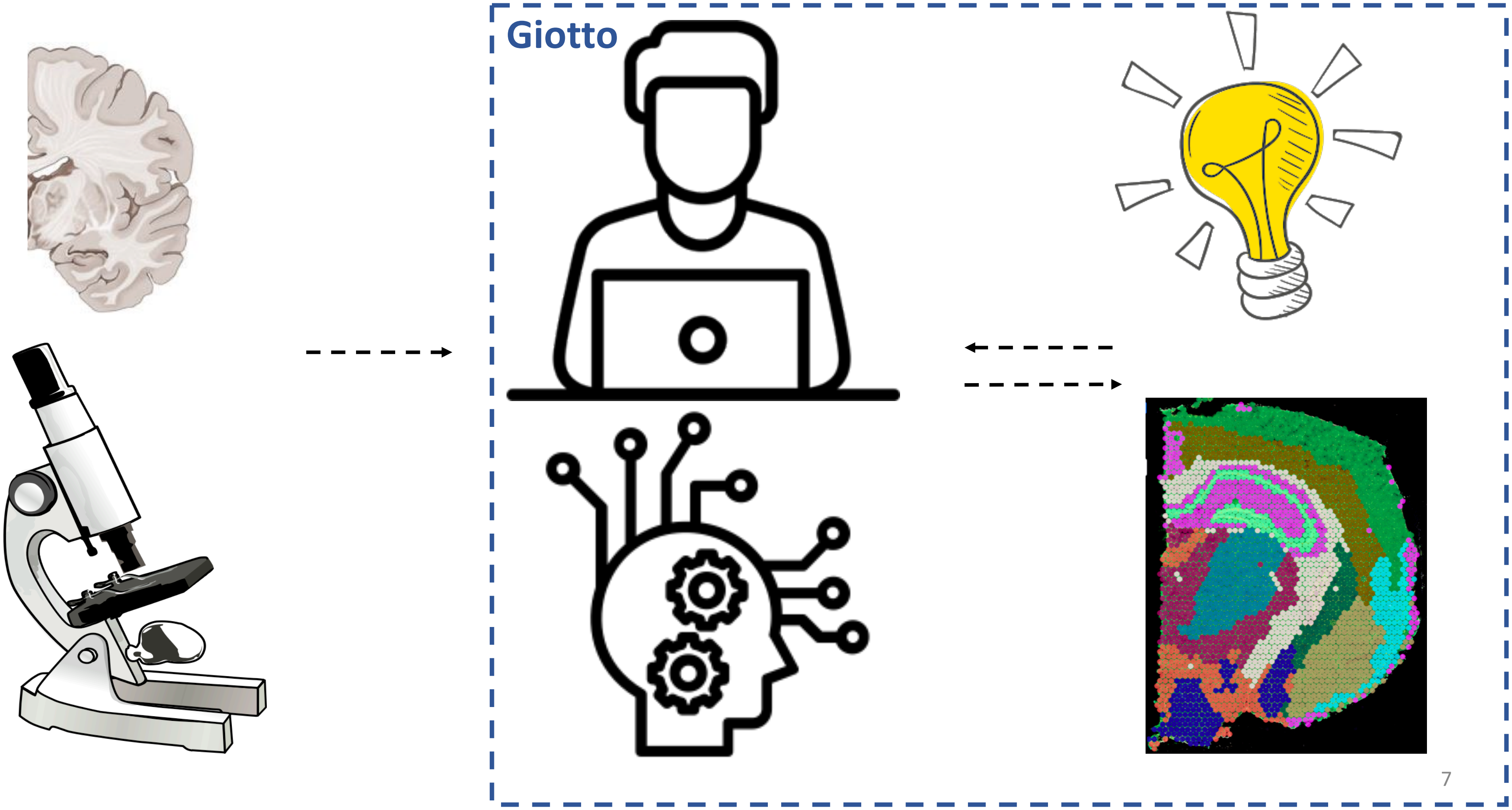
https://rubd.github.io/Giotto_site

- Start tab
- FAQ tab

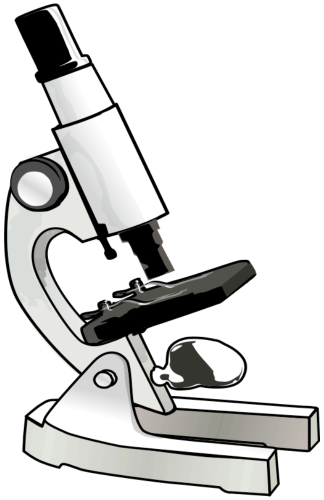
# Generating insight from spatial data

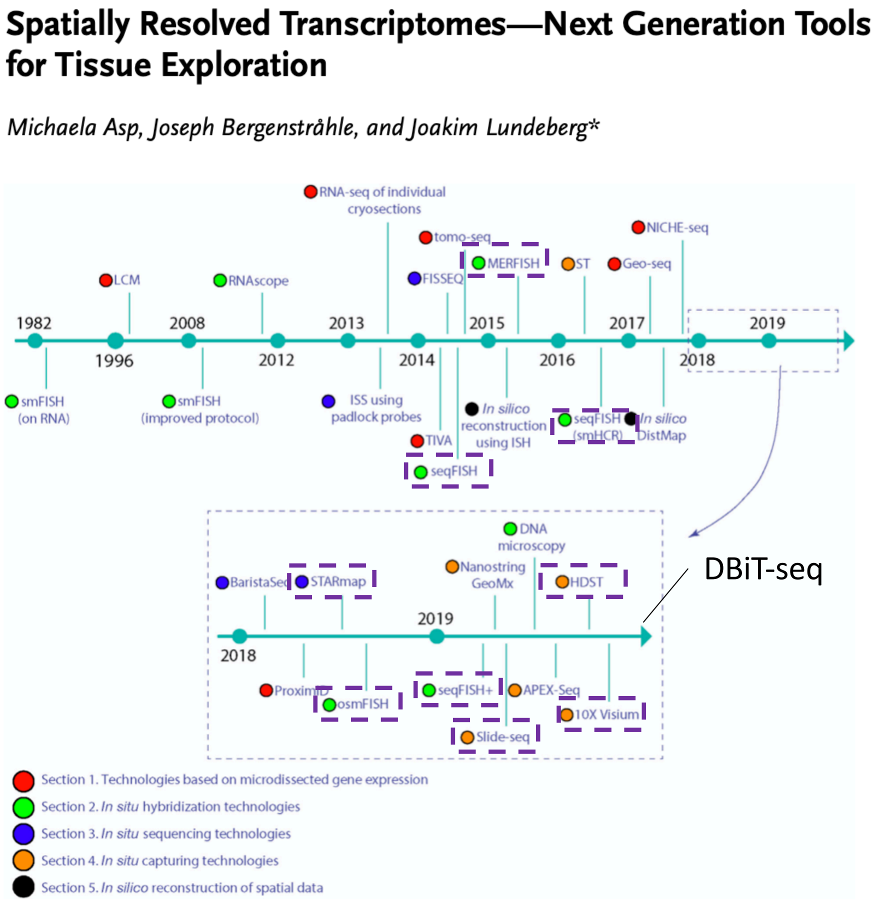


# Generating insight from spatial data



# Generating insight from spatial data



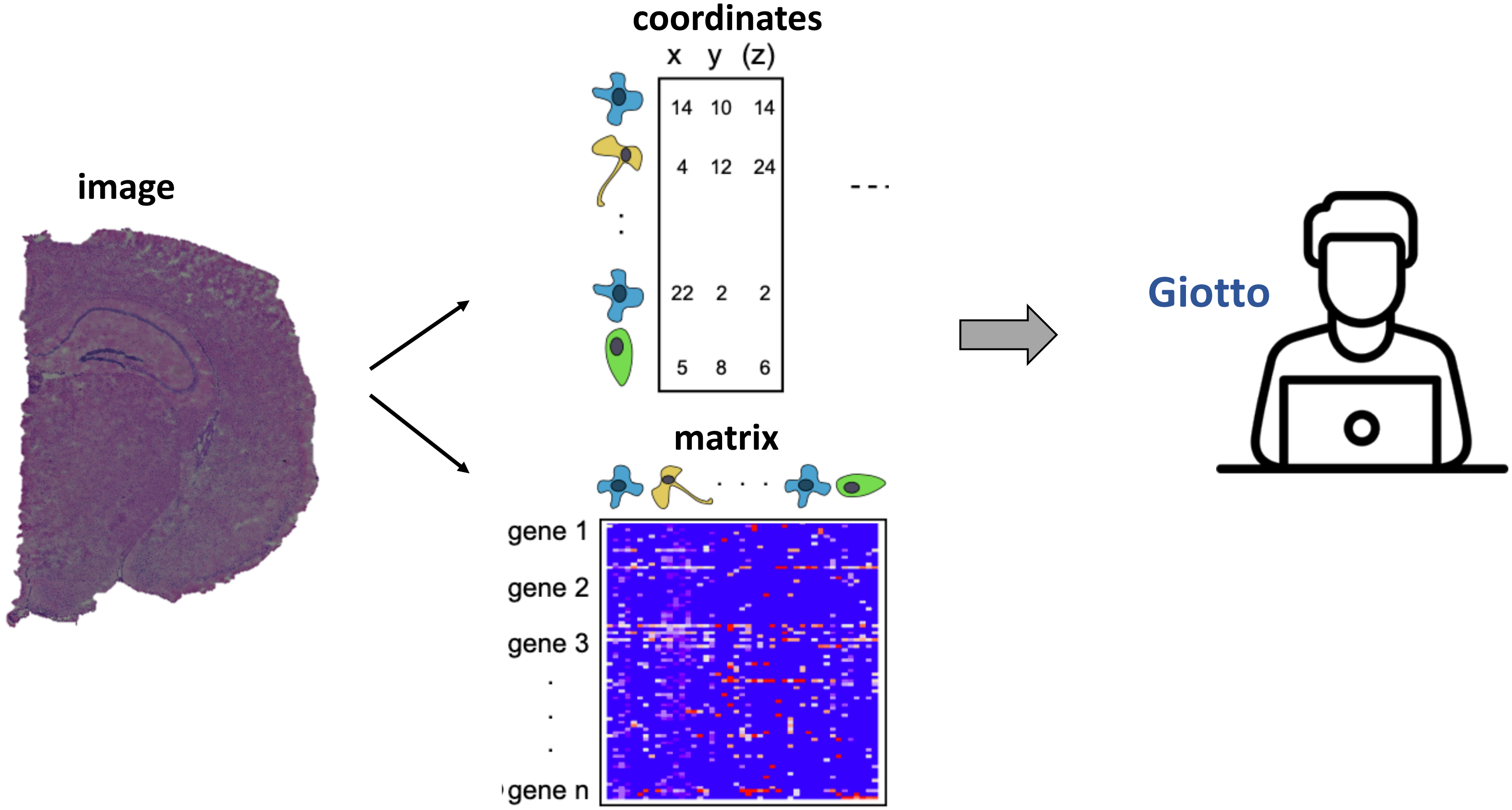


## Spatially Resolved Transcriptomes—Next Generation Tools for Tissue Exploration

*Michaela Asp, Joseph Bergenstråhle, and Joakim Lundeberg**

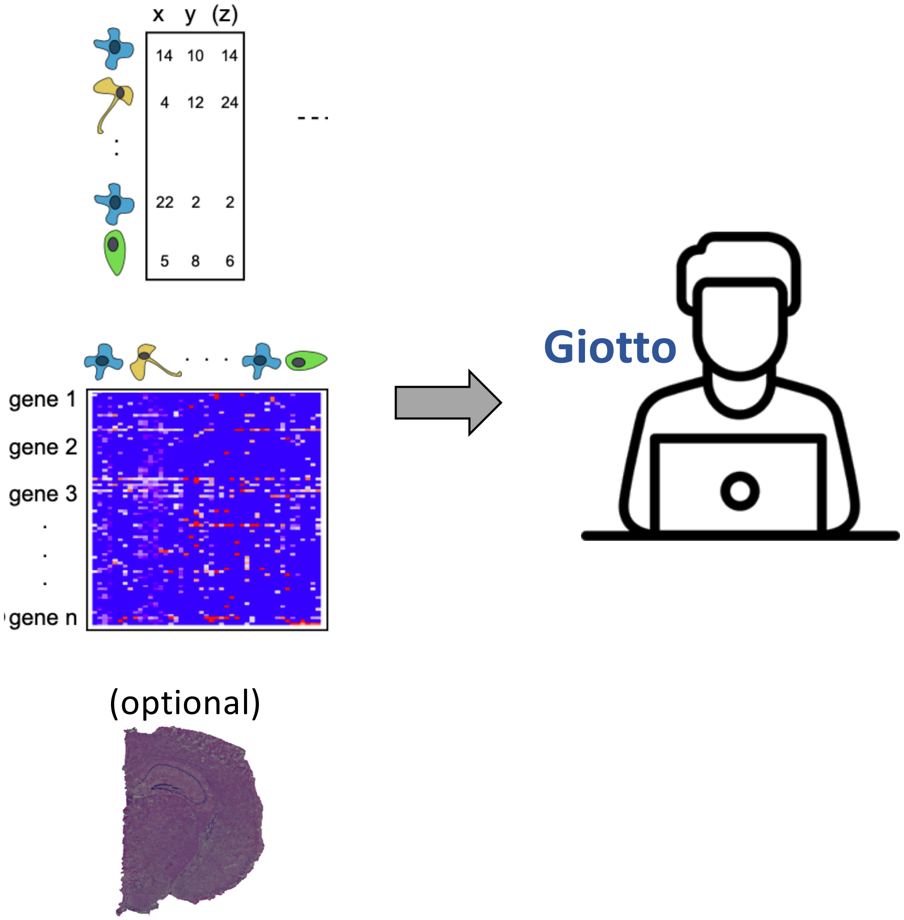
- 🔴 Section 1. Technologies based on microdissected gene expression
- 🟢 Section 2. *In situ* hybridization technologies
- 🔵 Section 3. *In situ* sequencing technologies
- 🟠 Section 4. *In situ* capturing technologies
- ⚫ Section 5. *In silico* reconstruction of spatial data

# Overview of Giotto analysis: input



| x | y | (z) |
|---|---|---|
| 14 | 10 | 14 |
| 4 | 12 | 24 |
| 22 | 2 | 2 |
| 5 | 8 | 6 |

# Overview of Giotto analysis: Giotto object



| x | y | (z) |
|---|---|---|
| 14 | 10 | 14 |
| 4 | 12 | 24 |
| 22 | 2 | 2 |
| 5 | 8 | 6 |

general:

```
myGobject = createGiottoObject()
```

```
myGimage = createGiottoImage()
```

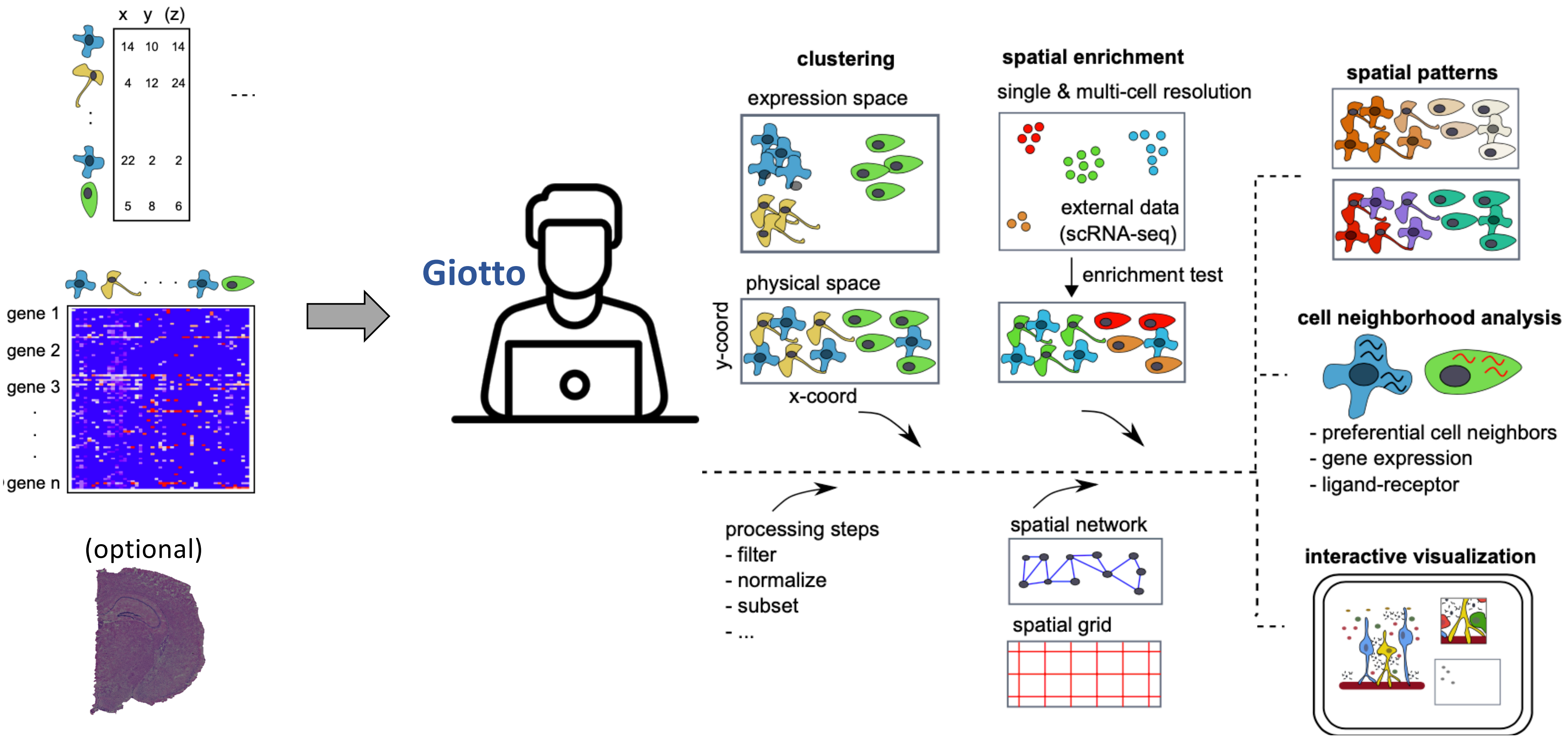
```
myGobject = addGiottoImage()
```

https://rubd.github.io/Giotto_site
- Start tab: Howtos (working with images)

Visium 10X (combined):

```
myGobject = createGiottoVisiumObject()
```

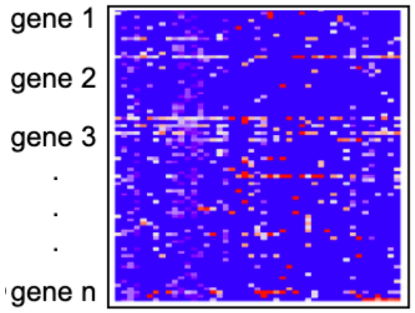
# Overview of Giotto analysis: interactive visualization



| x | y | (z) |
|---|---|---|
| 14 | 10 | 14 |
| 4 | 12 | 24 |
| 22 | 2 | 2 |
| 5 | 8 | 6 |

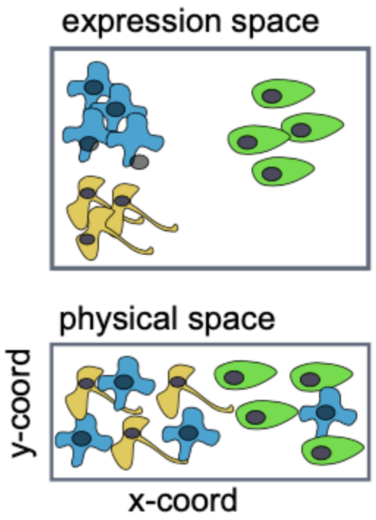
# Cell type analyses

Input:



| x | y | (z) |
|---|---|---|
| 14 | 10 | 14 |
| 4 | 12 | 24 |
| 22 | 2 | 2 |
| 5 | 8 | 6 |

Output:

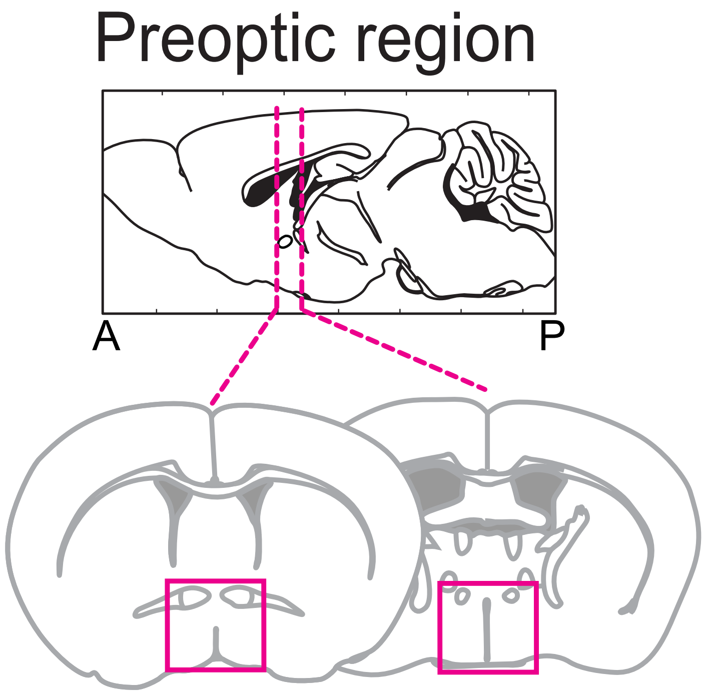


Giotto:

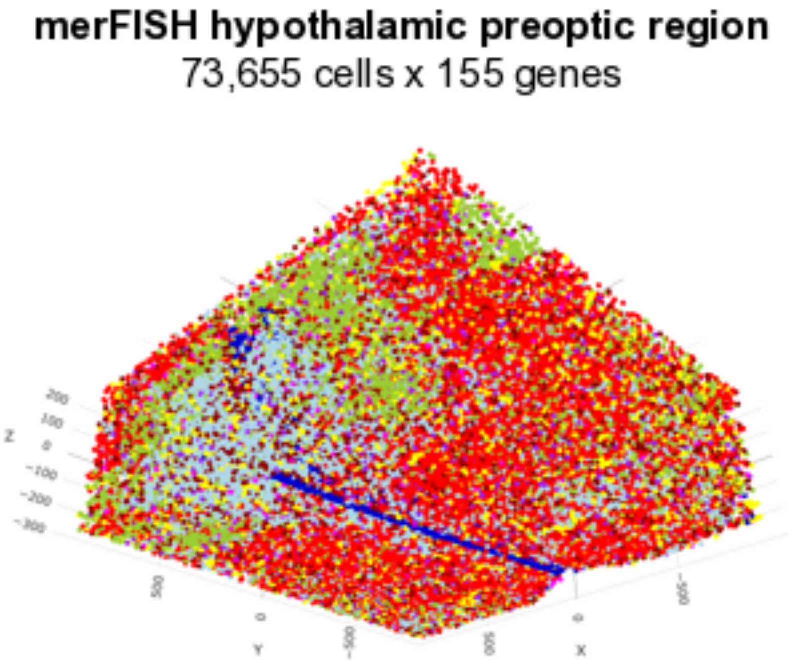
- createGiottoObject
- filterGiotto
- normalizeGiotto
- runPCA
- createNearestNetwork
- doLeidenCluster
- findMarkers
- annotateGiotto
- …

# Cell type analyses: MERFISH

Example: mouse hypothalamic preoptic regions (MERFISH)



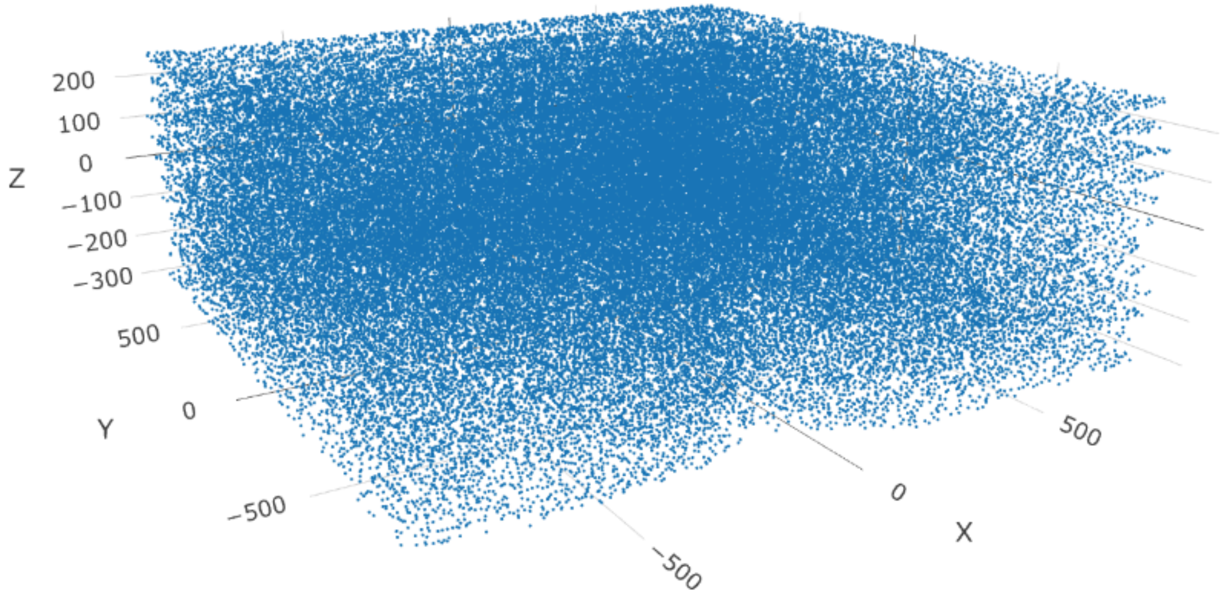
Preoptic region



merFISH hypothalamic preoptic region
73,655 cells x 155 genes

# Cell type analyses: create object

```
myGobject = createGiottoObject (expr, location)
```



# Cell type analyses: dimension reduction

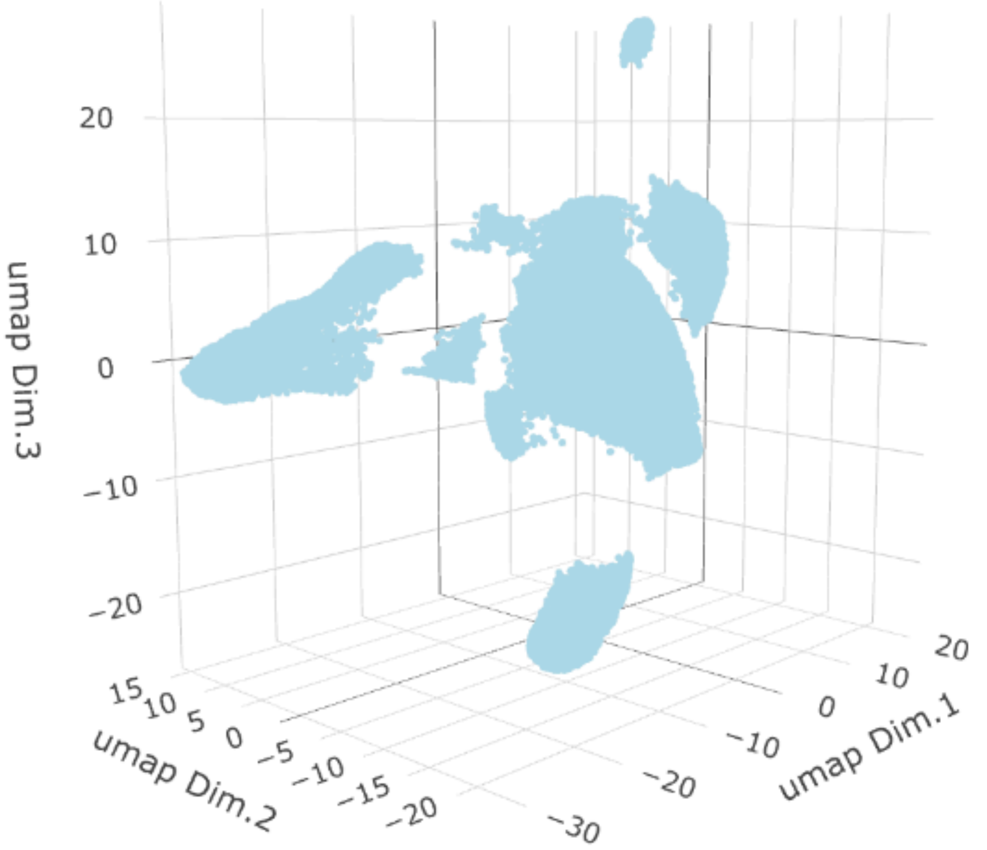
```
myGobject = filterGiotto (myGobject, …)
```

```
myGobject = normalizeGiotto (myGobject, …)
```

```
myGobject = runPCA (myGobject, …)
```

```
myGobject = runUMAP (myGobject, …)
```

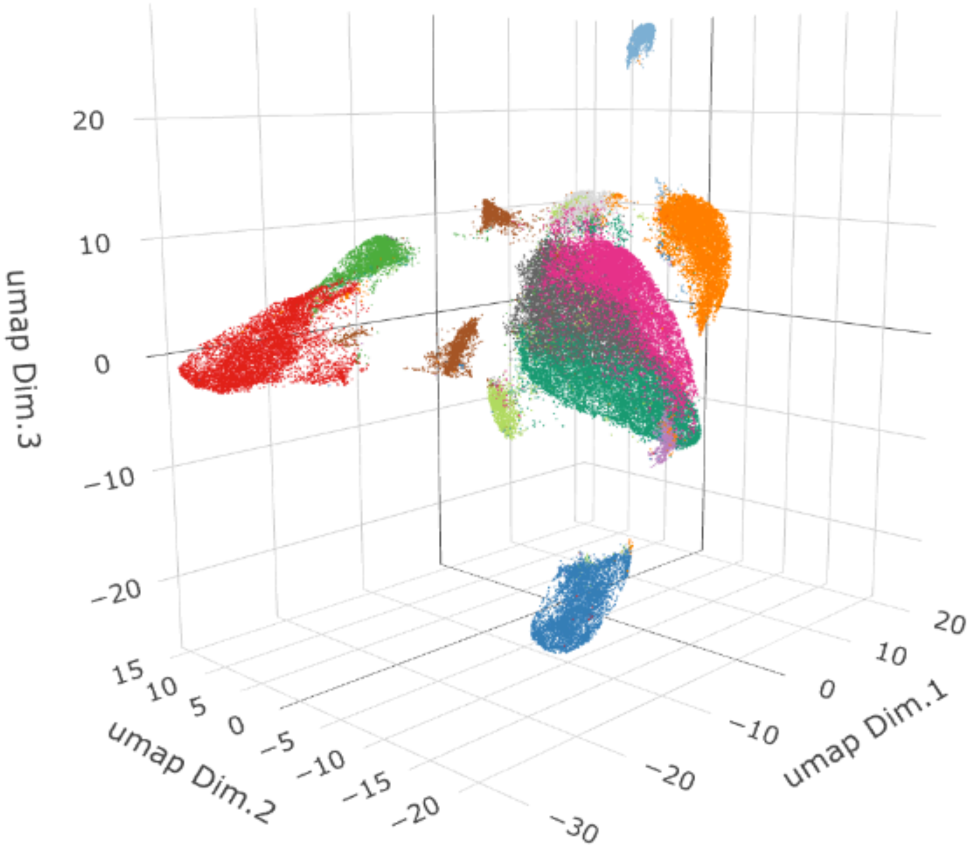
```
plotUMAP_3D (myGobject, …)
```



# Cell type analyses: clustering

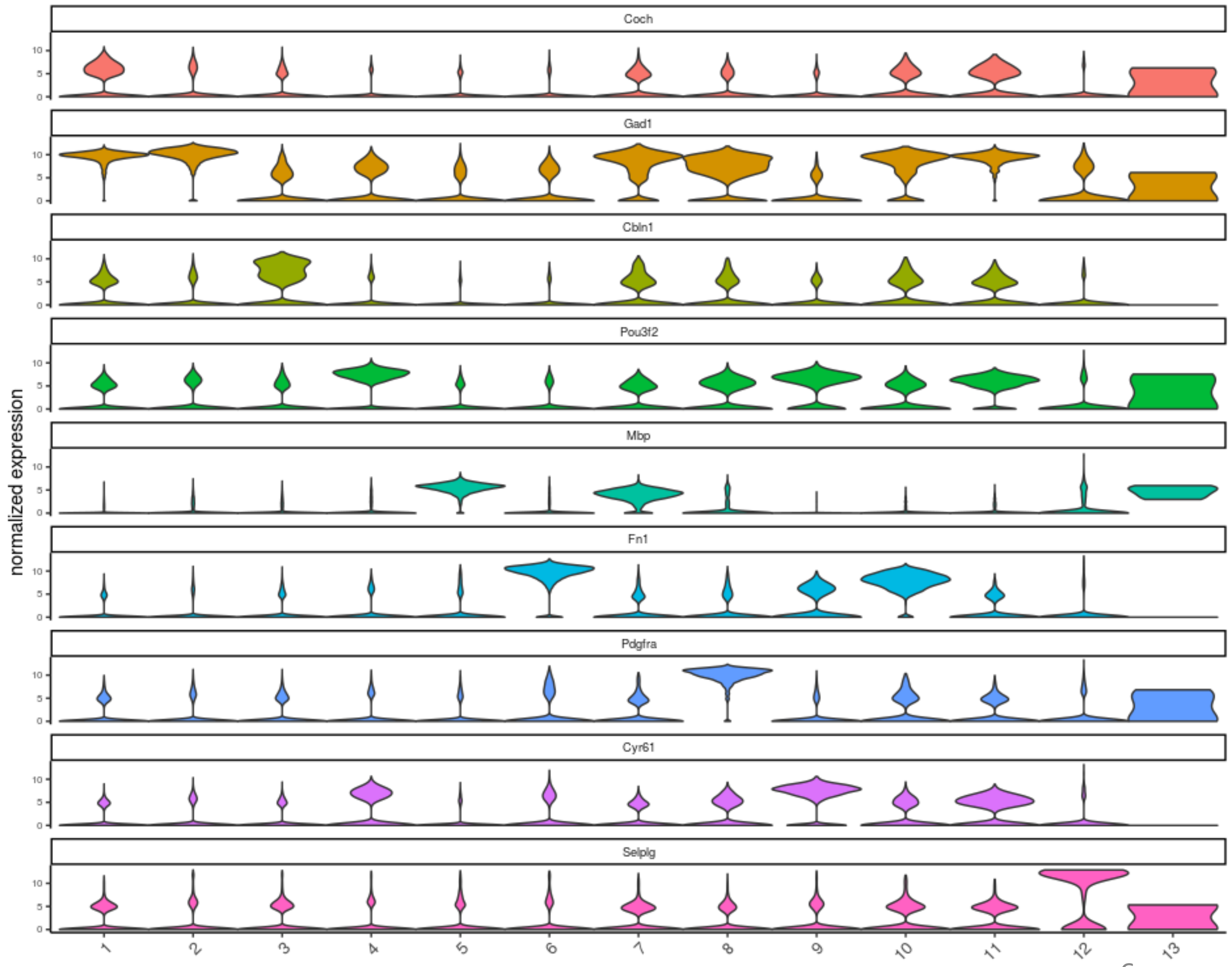
```
myGobject = createNearestNetwork (myGobject, …)
```

```
myGobject = doLeidenCluster (myGobject, …)
```



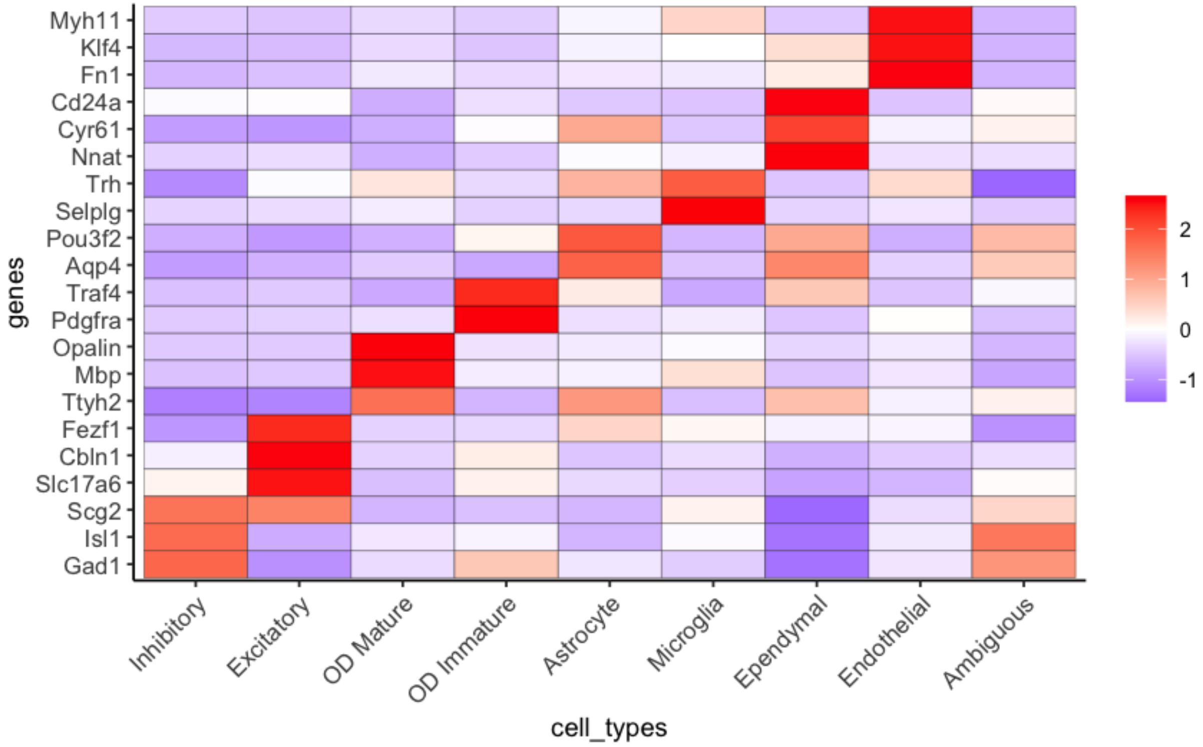
# Cell type analyses: find markers

```
myGobject = findMarkers (myGobject, …)
```



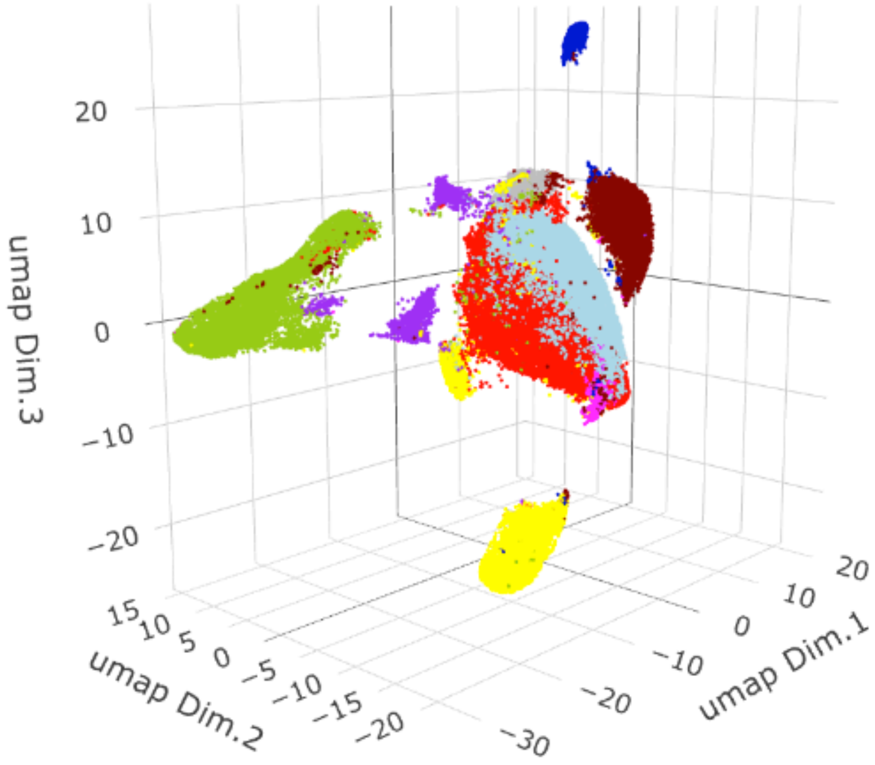
# Cell type analyses: annotation

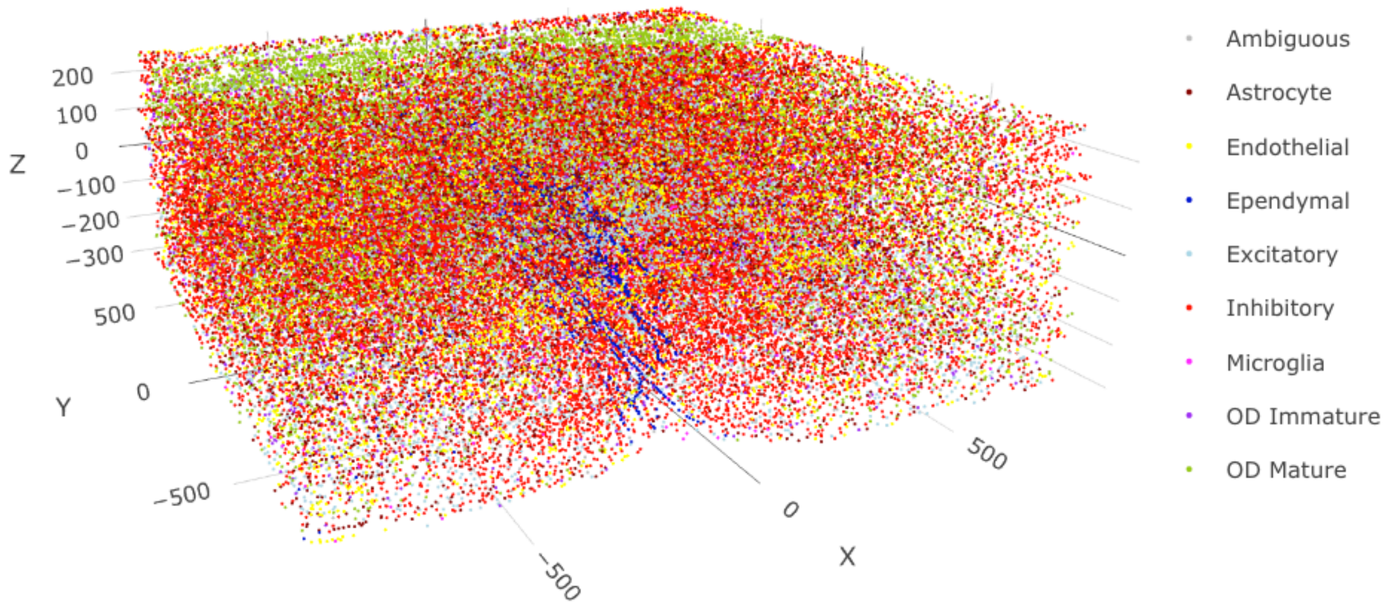
```
myGobject = annotateGiotto (myGobject, …)
```



# Cell type analyses: annotation

```
spatDimPlot3D (myGobject, …)
```

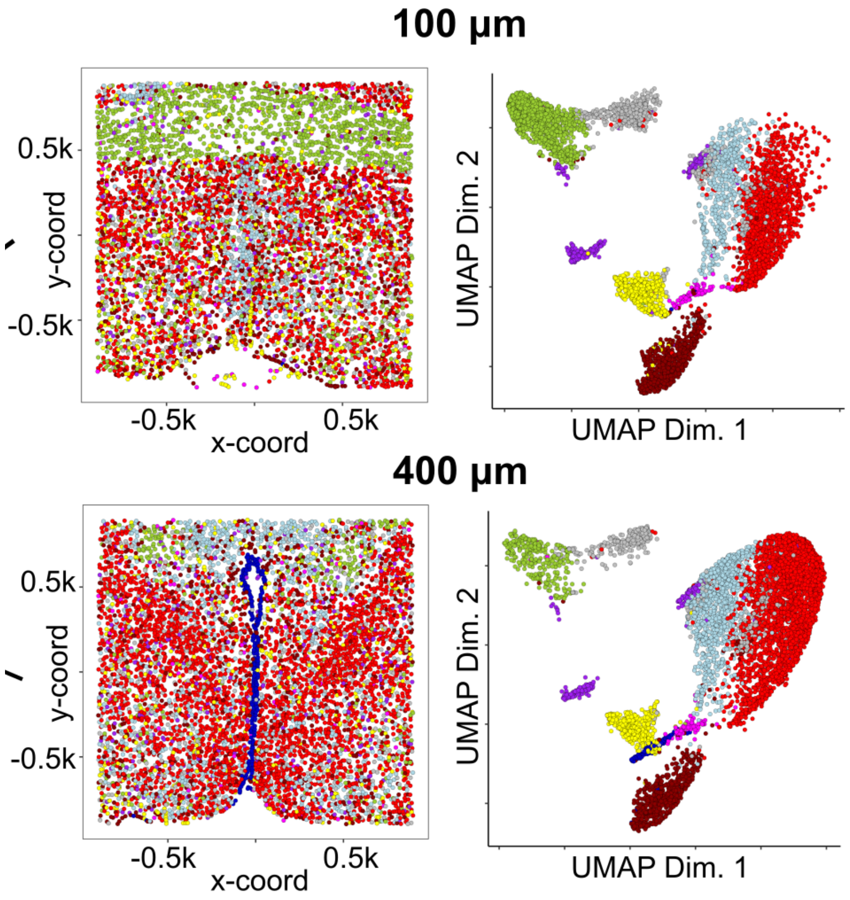


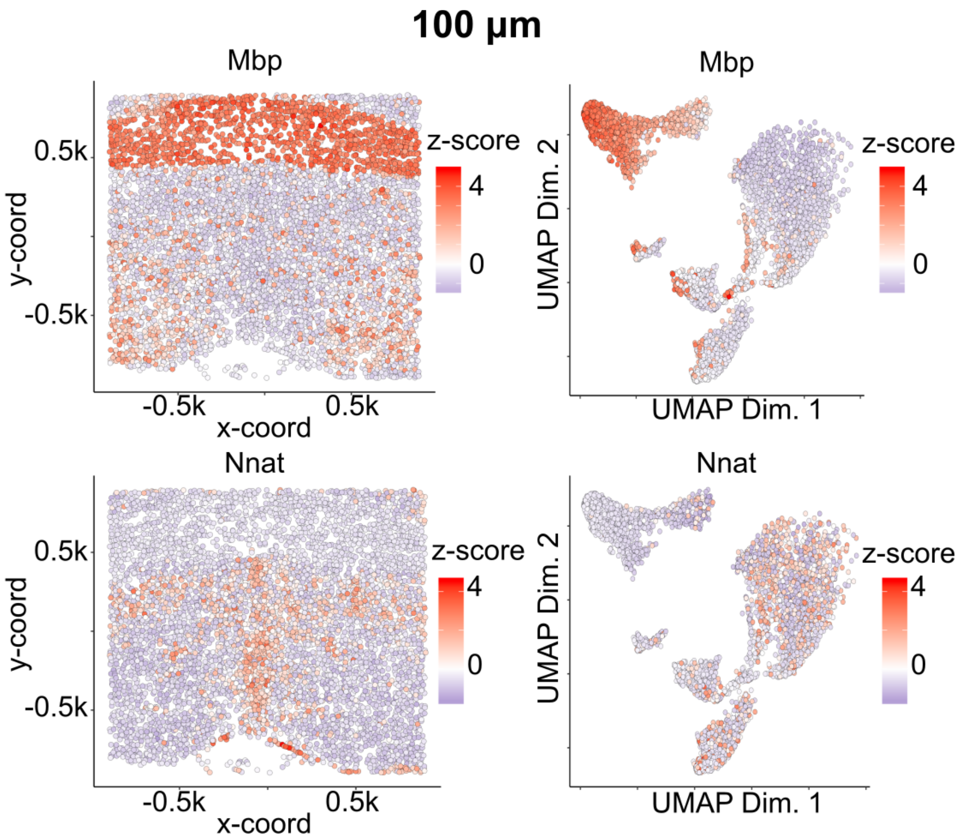


# Cell type analyses: 2D slices

spatDimPlot (myGobject, …)

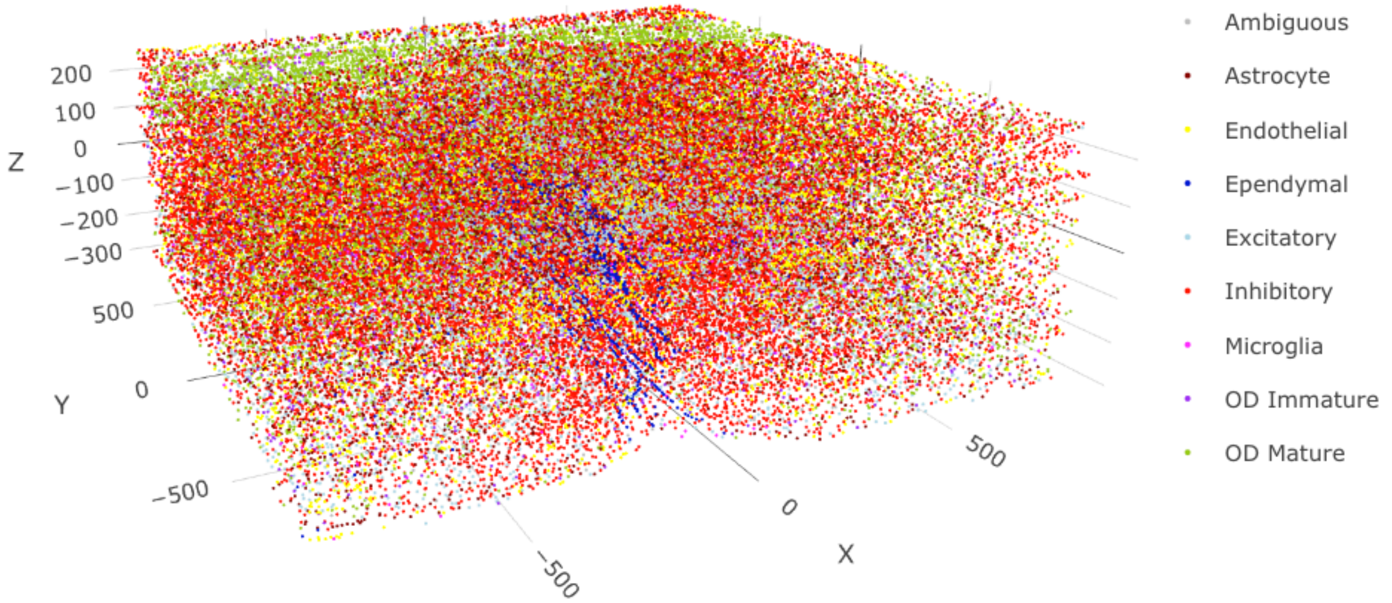
spatDimGenePlot (myGobject, …)



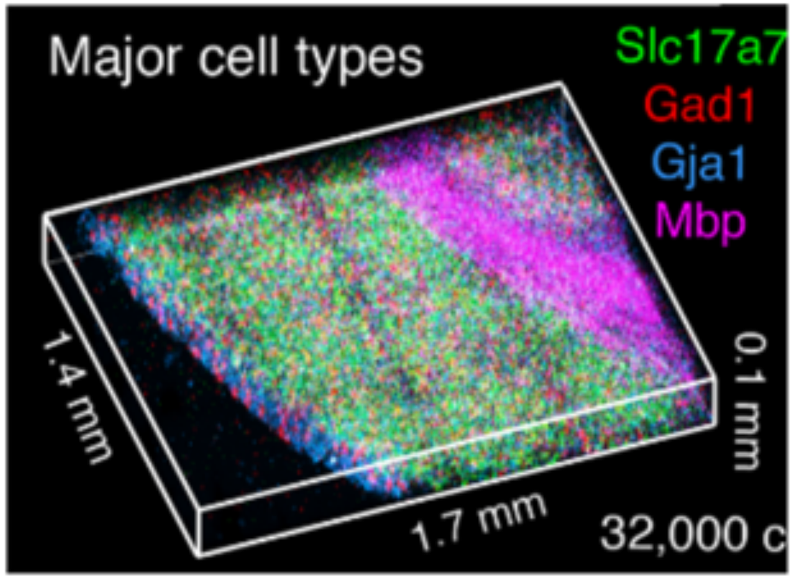


# Cell type analyses: virtual 2D slices

MERFISH



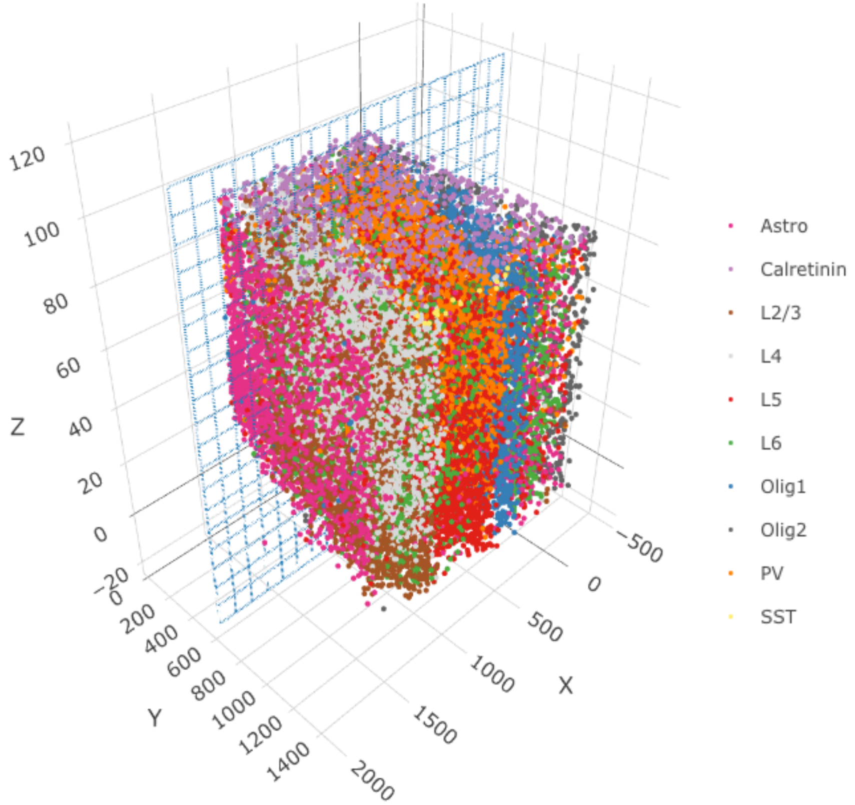
**STARmap data (3D)**
32,841 cells x 28 genes

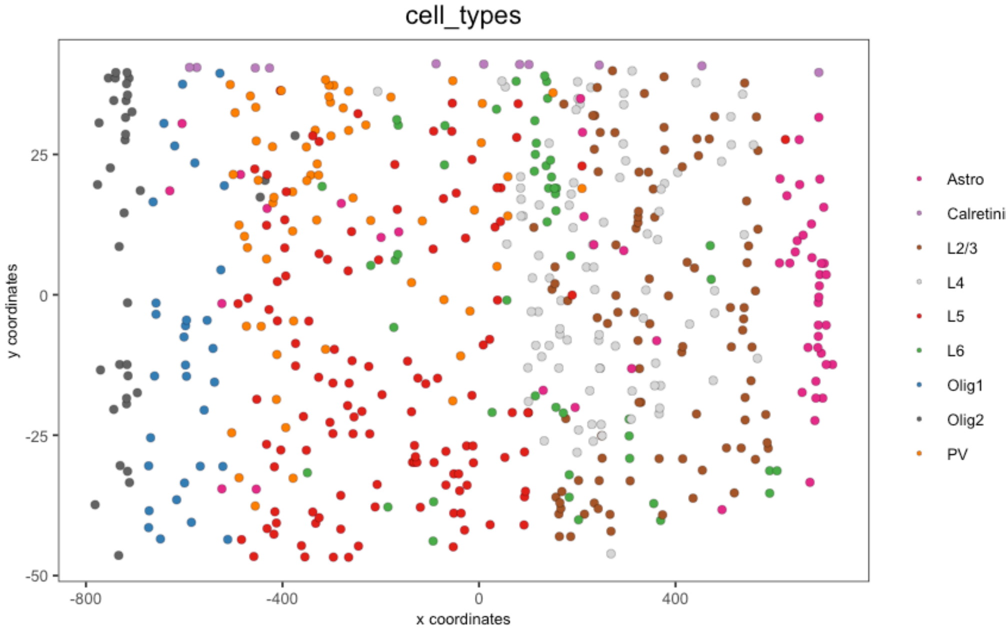


# Cell type analyses: virtual 2D slices

```
myGobject = createCrossSection (myGobject, …)
```

```
crossSectionPlot (myGobject, …)
```

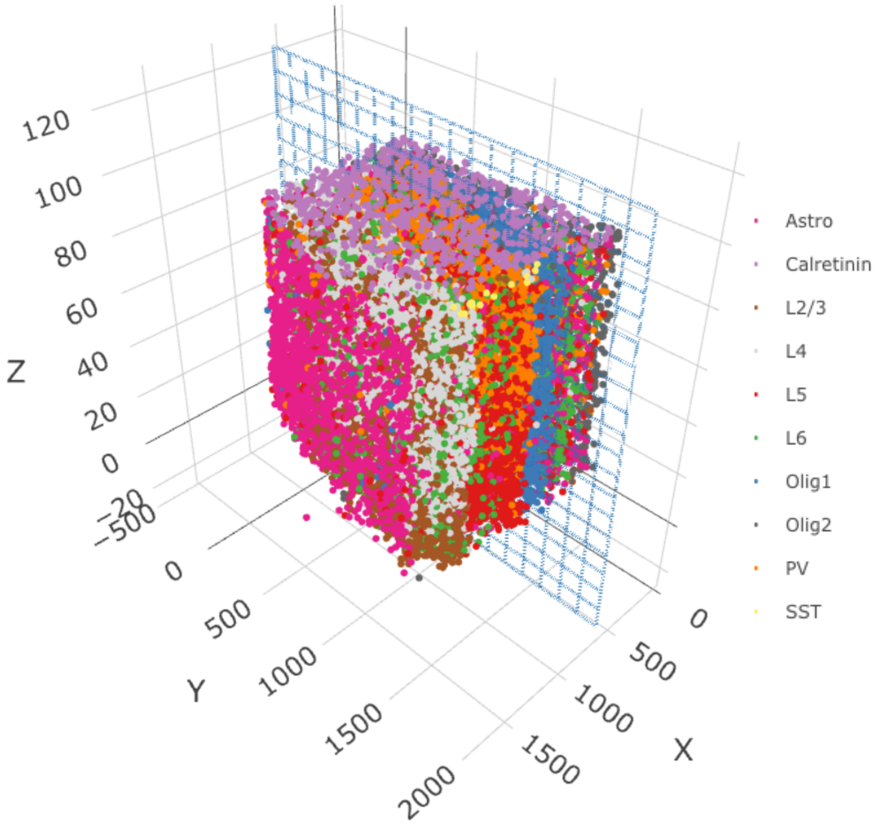


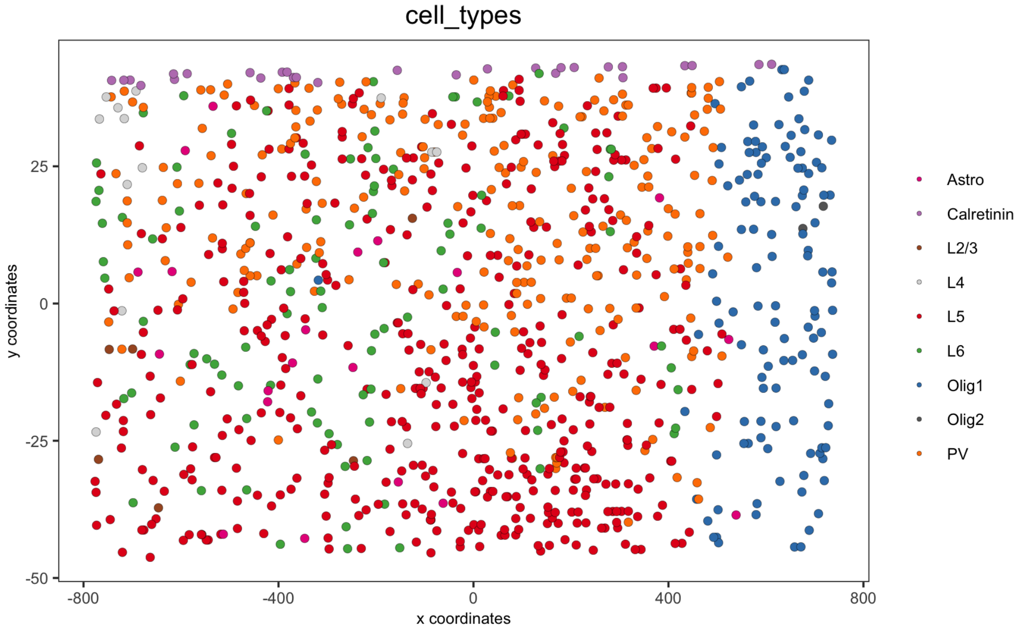


# Cell type analyses: virtual 2D slices

```
myGobject = createCrossSection (myGobject, …)
```

```
crossSectionPlot (myGobject, …)
```





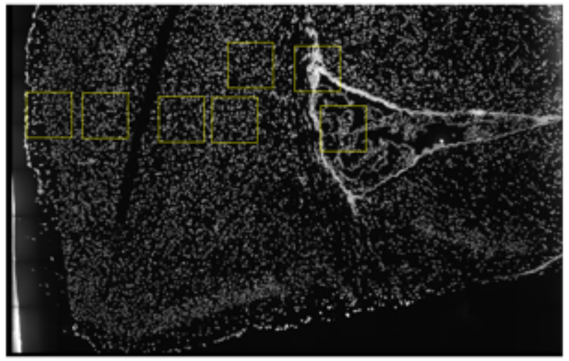
# Cell type enrichment analysis for lower-resolution spatial expression datasets

Rui Dong
06/11/2020

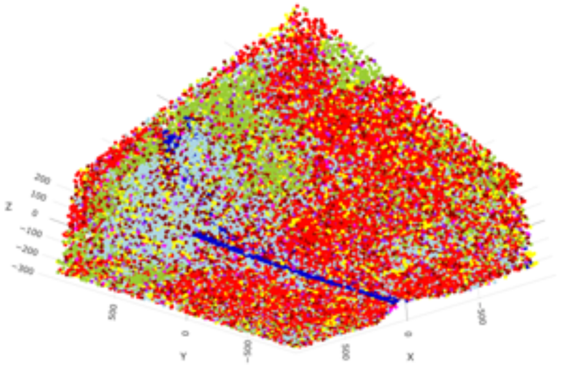
# From single cell to lower-resolution spatial expression datasets

## Single cell spatial expression datasets

seqFISH+

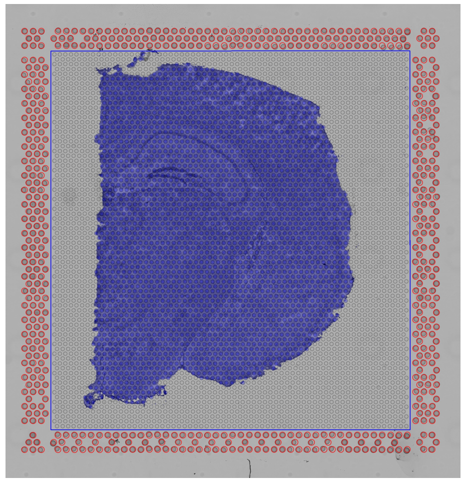


merFISH

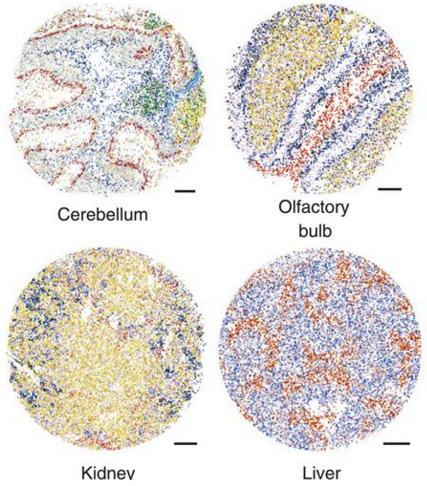


## Lower-resolution spatial expression datasets

Visium



Slide-seq



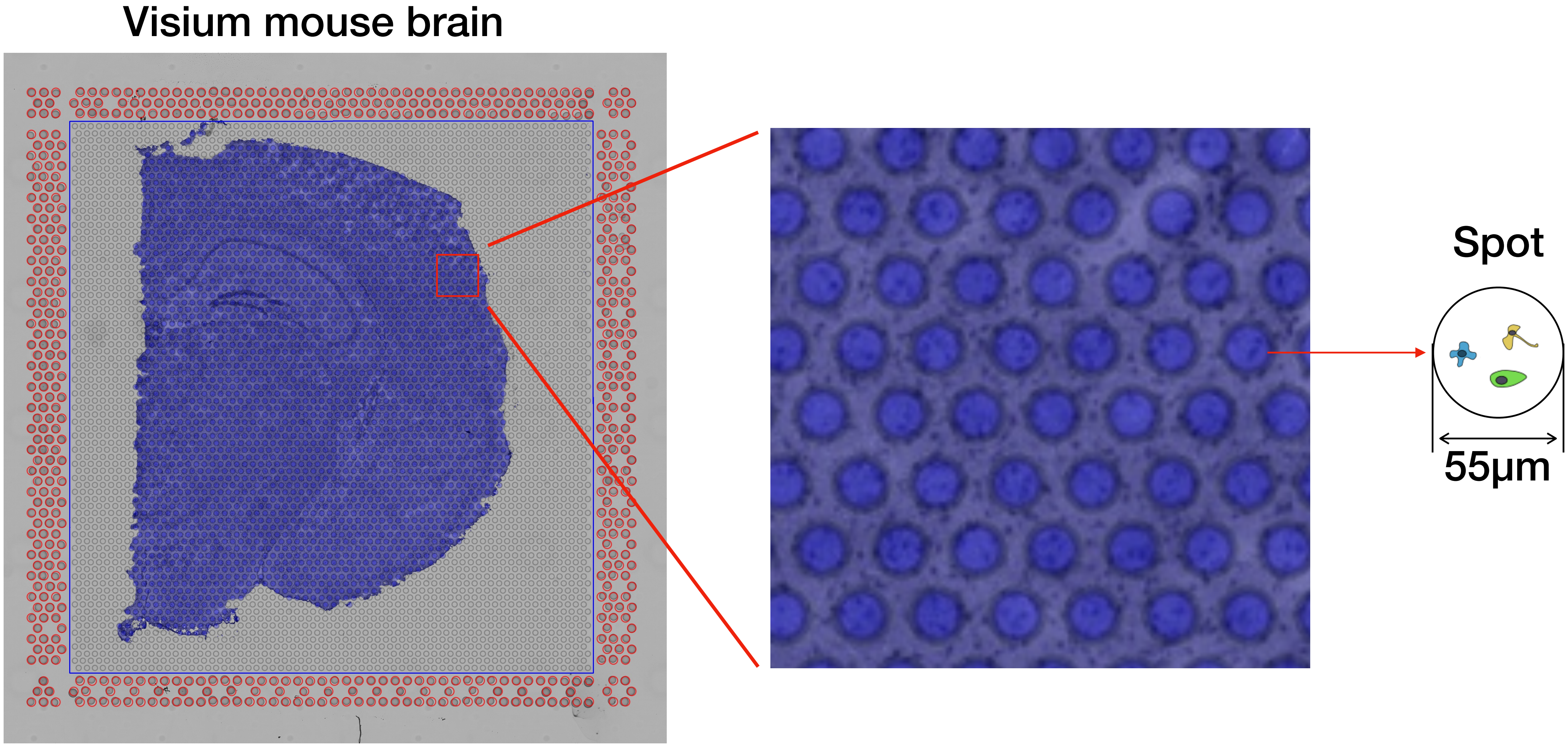
Cerebellum

Olfactory bulb

Kidney

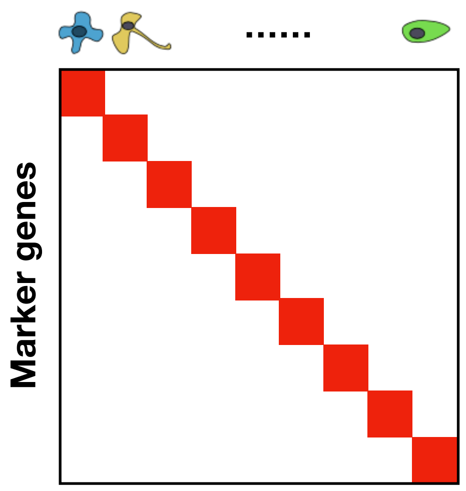
Liver

# Multiple cells may contained in one spot

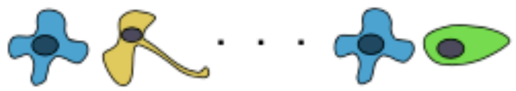


# Enrichment pipeline for PAGE and RANK

PAGE (Parametric Analysis of Gene set Enrichment)



RANK



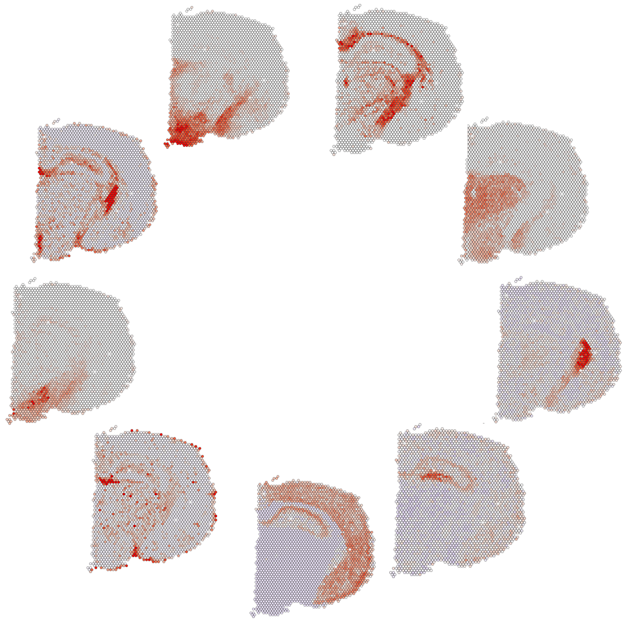
Cell type labels

| | Cell1 | Cell2 | ...... | Cell$_{n-1}$ | Cell$_n$ |
|---|---|---|---|---|---|
| Gene1 | | | | | |
| Gene2 | | | | | |
| Gene3 | | | | | |
| Gene4 | | | | | |
| ...... | | | | | |

scRNA-seq expression matrix

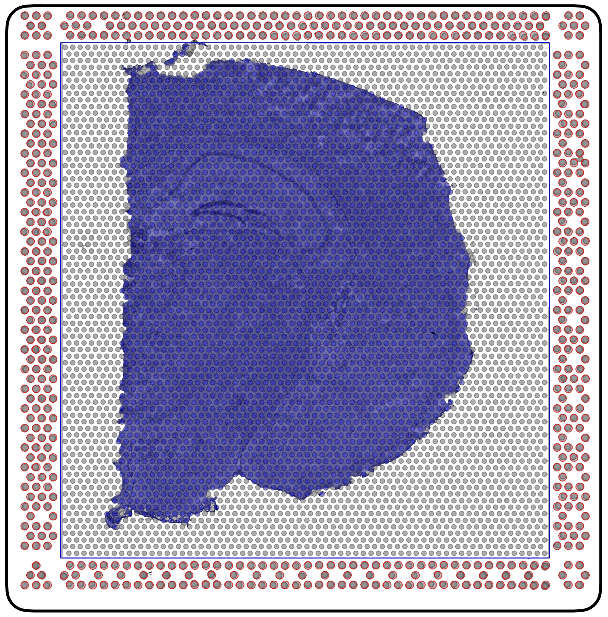
Giotto enrichment

Enrichment results

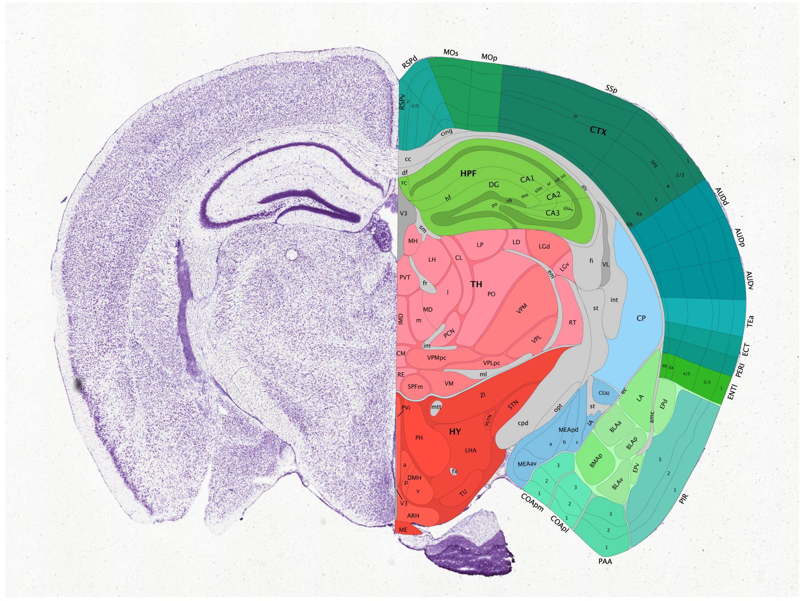


# Enrichment analysis of low-resolution spatial expression dataset

Visium mouse brain dataset from 10X



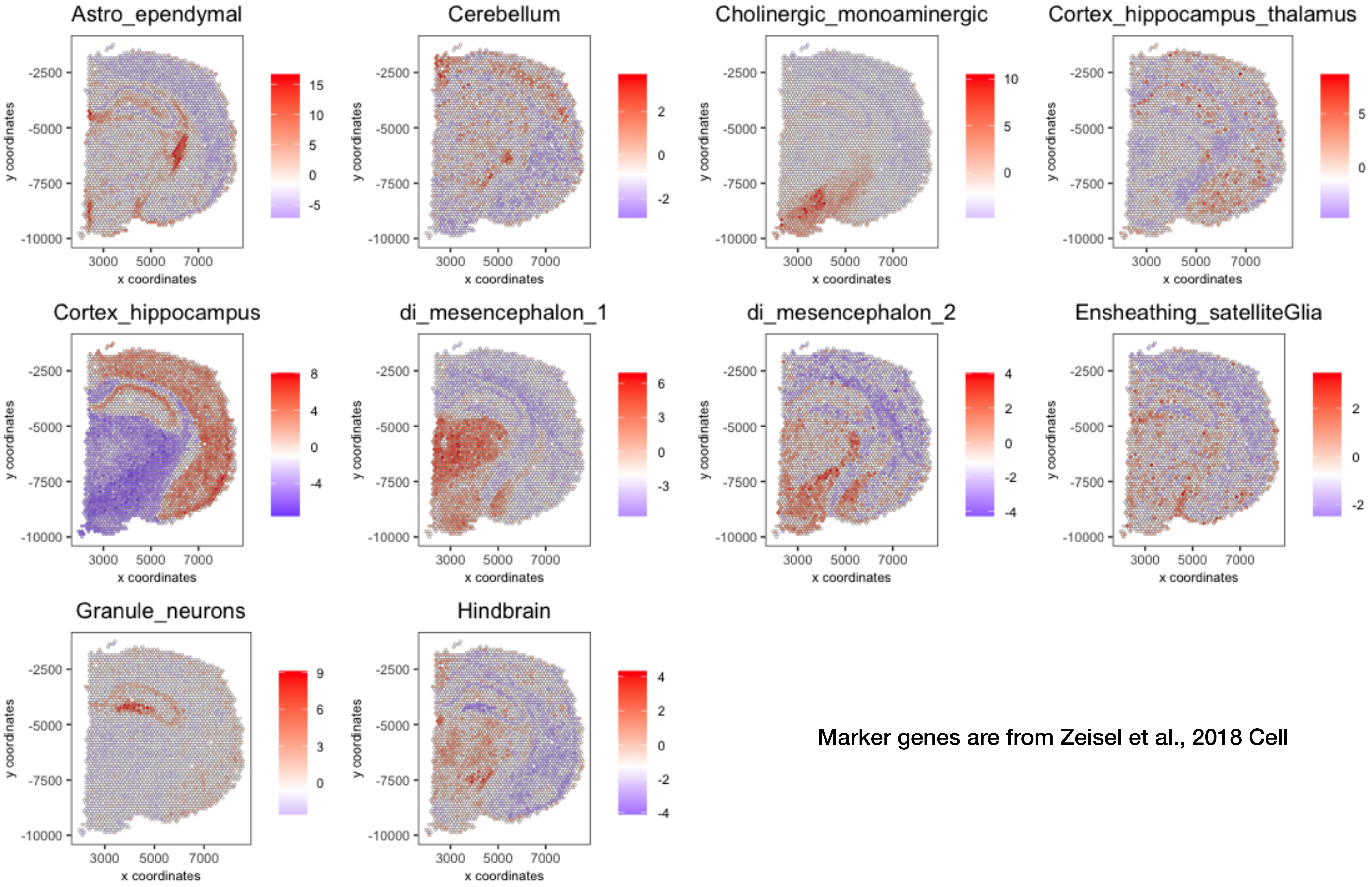
Coronal mouse brain from Allen Brain Atlas



Example: mouse brain (Visium 10X)

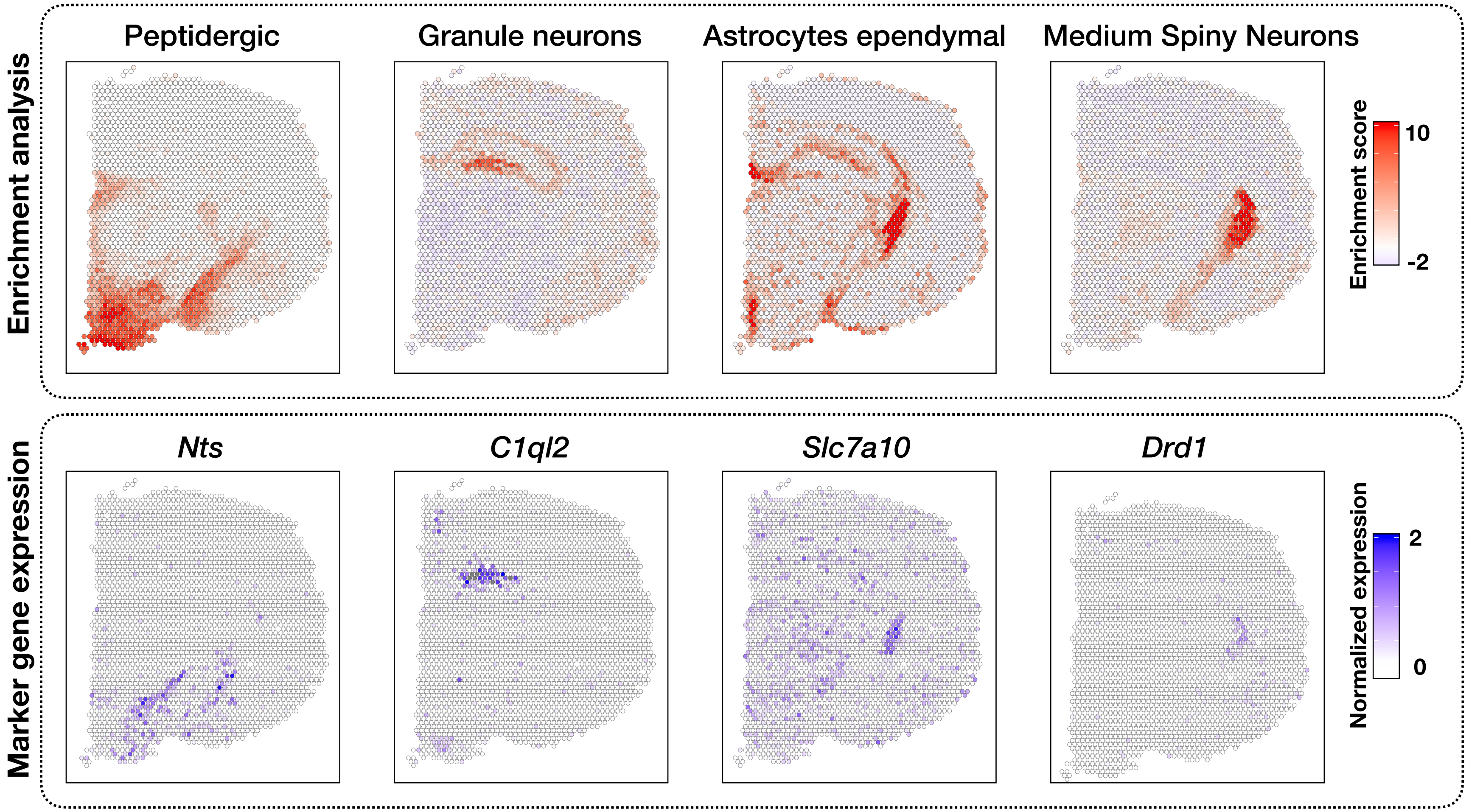
# Enrichment analysis and visualization using PAGE

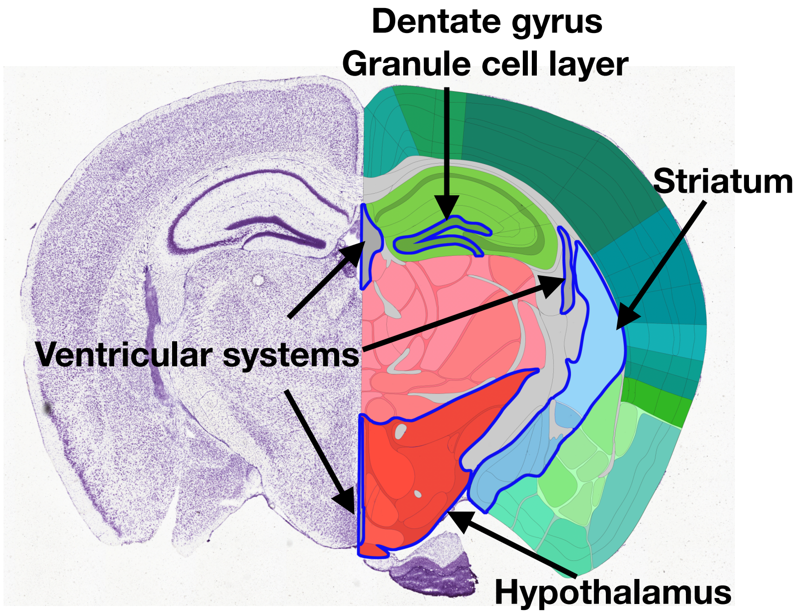
```
myGobject = createSpatialEnrich()
spatCellPlot()
```



Marker genes are from Zeisel et al., 2018 Cell

# Consistency between cell type enrichment and Allen Brain Atlas



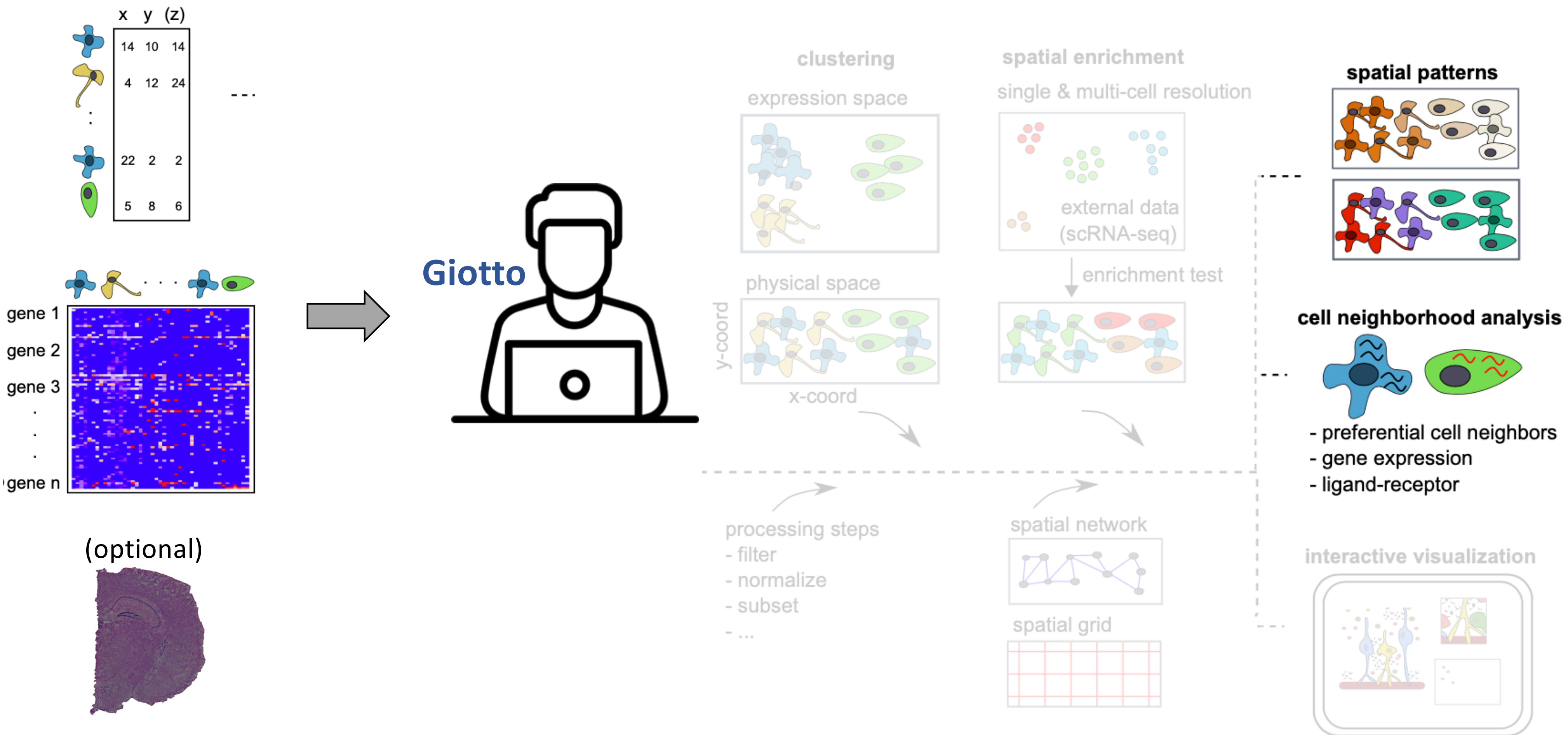


# Giotto spatial analyses

Spatial analyses

**Ruben Dries**

# Spatial analyses



| x | y | (z) |
|---|---|---|
| 14 | 10 | 14 |
| 4 | 12 | 24 |
| 22 | 2 | 2 |
| 5 | 8 | 6 |

gene 1
gene 2
gene 3
gene n

(optional)

Giotto

**clustering**

expression space

physical space

y-coord

x-coord

**spatial enrichment**

single & multi-cell resolution

external data (scRNA-seq)

enrichment test

processing steps
- filter
- normalize
- subset
- ...

spatial network

spatial grid

**spatial patterns**

**cell neighborhood analysis**

- preferential cell neighbors
- gene expression
- ligand-receptor

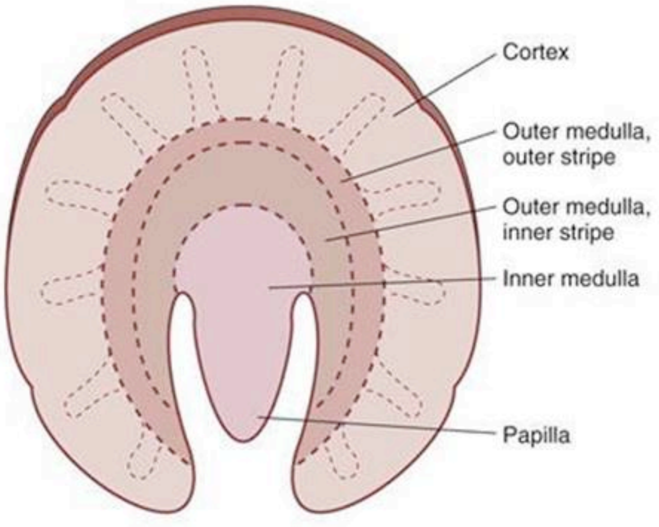
**interactive visualization**

# Spatial analyses

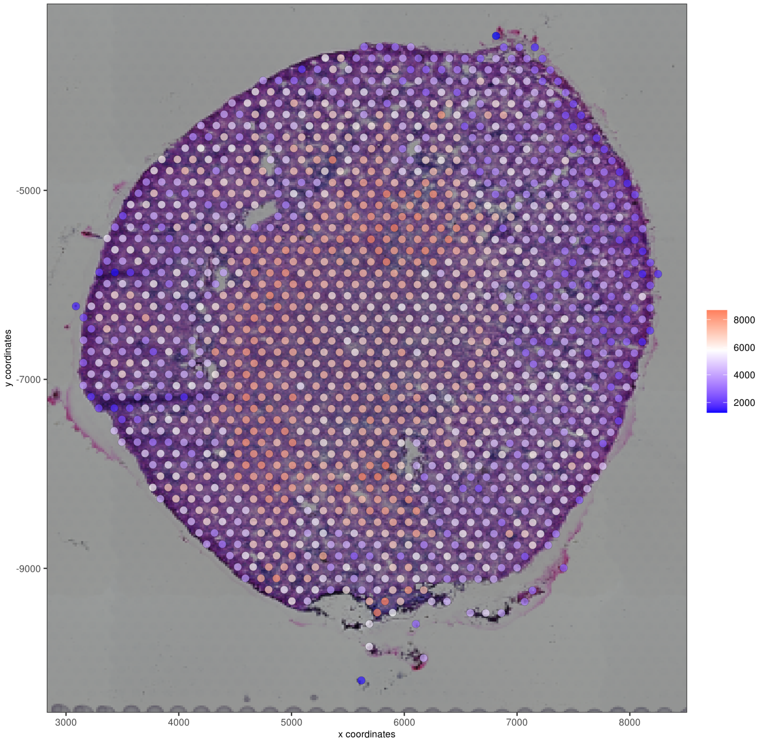
1. Spatial patterns:
    1. individual genes / proteins
    2. continuous spatial expression modules
    3. discrete spatial domains

# Spatial analyses: kidney organization

Example: mouse kidney (Visium 10X)



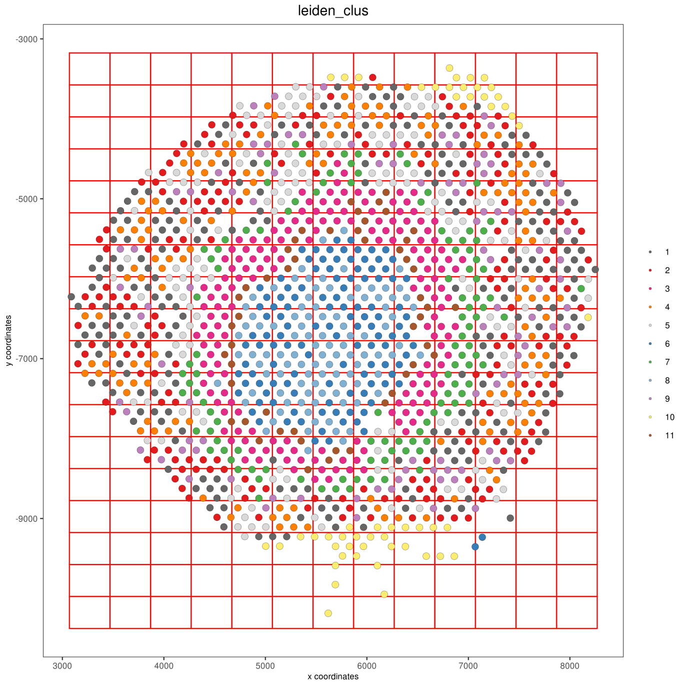
# genes detected per spot

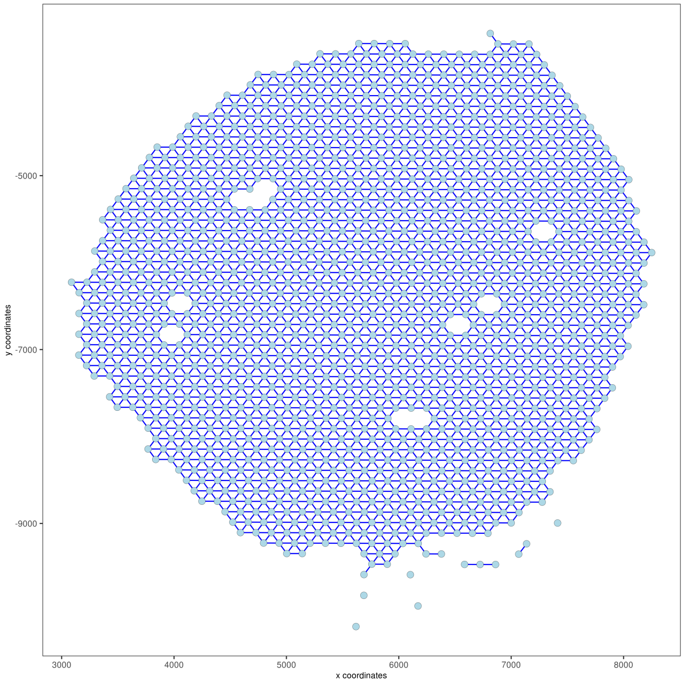


# Spatial analyses: spatial grid and/or network

```
myGobject = createSpatialGrid(myGobject)
```

```
myGobject = createSpatialNetwork(myGobject)
```

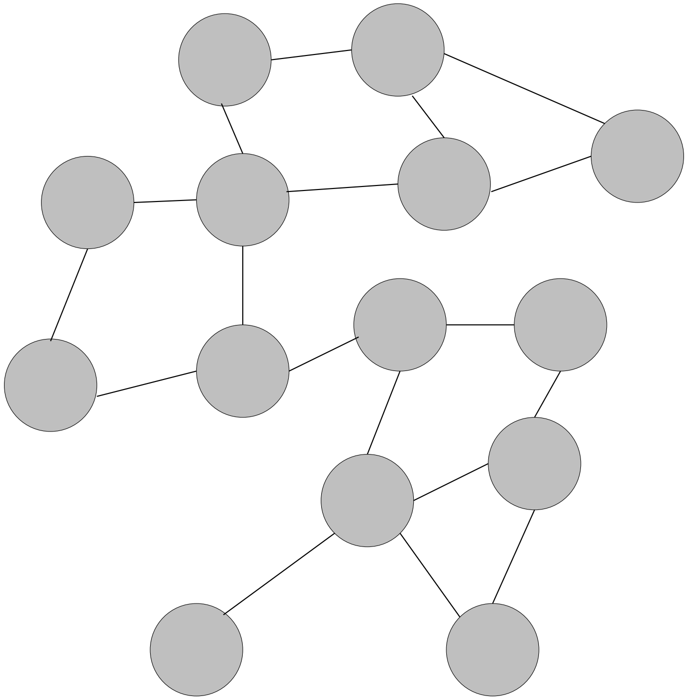




# Spatial analyses: spatial network

● = Cell

- k neighbors
- radial distance
- minimum
- Delaunay network

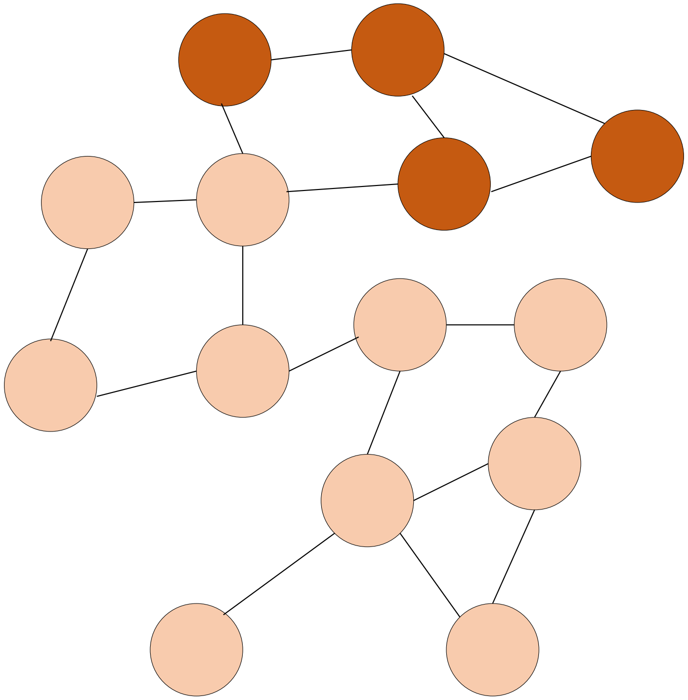


# Spatial analyses: spatial network

spatially coherent genes

● gene X **high**

○ gene X **low**



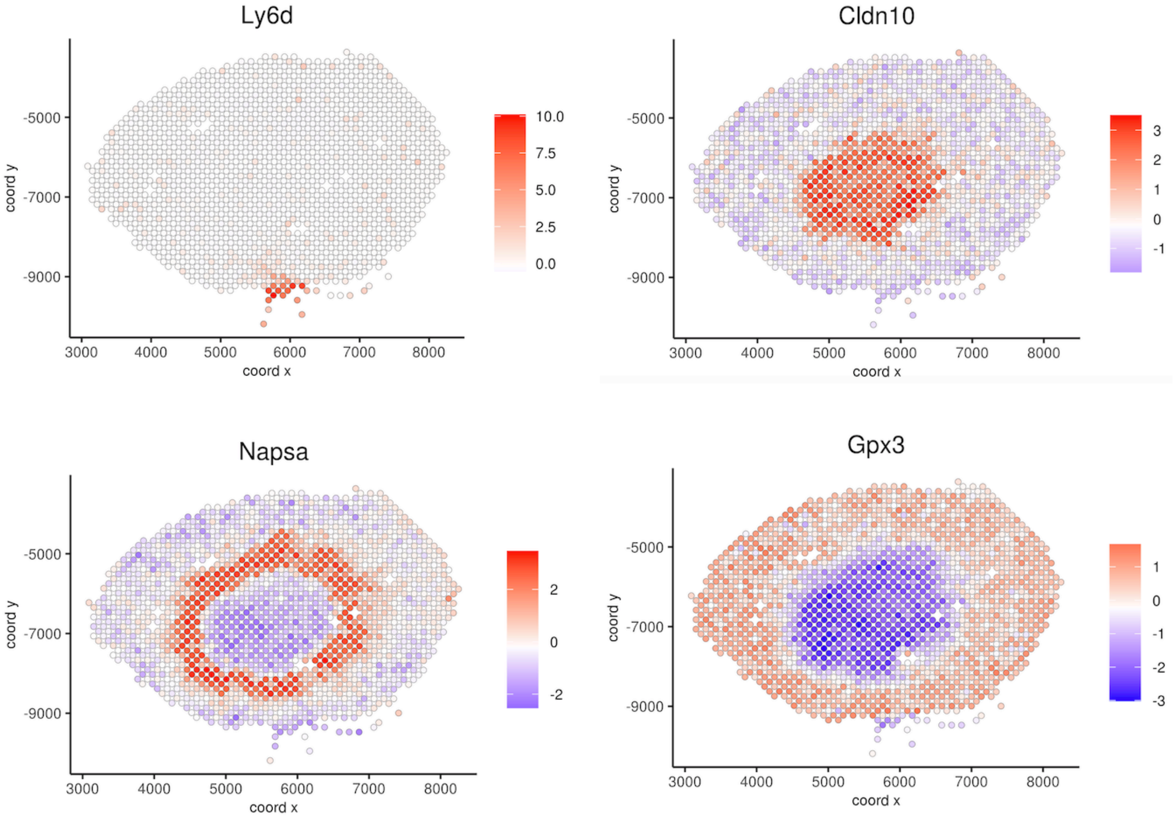
# Spatial analyses: spatial genes

`bin_genes = binSpect()`

`sil_genes = silhouetteRank()` [*1]

`spatialDE_genes = spatialDE()` [*2]

`trend_genes = trendSceek()` [*3]



1. Zhu, Q., Shah, S., Dries, R., Cai, L. & Yuan, G.-C. Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence *in situ* hybridization data. *Nature Biotechnology* (2018)
2. Svensson, V., Teichmann, S. A. & Stegle, O. SpatialDE: identification of spatially variable genes. *Nat Methods* **15**, 343–346 (2018).
3. Edsgärd, D., Johnsson, P. & Sandberg, R. Identification of spatial expression trends in single-cell gene expression data. *Nat Methods* **15**, 339–342 (2018).

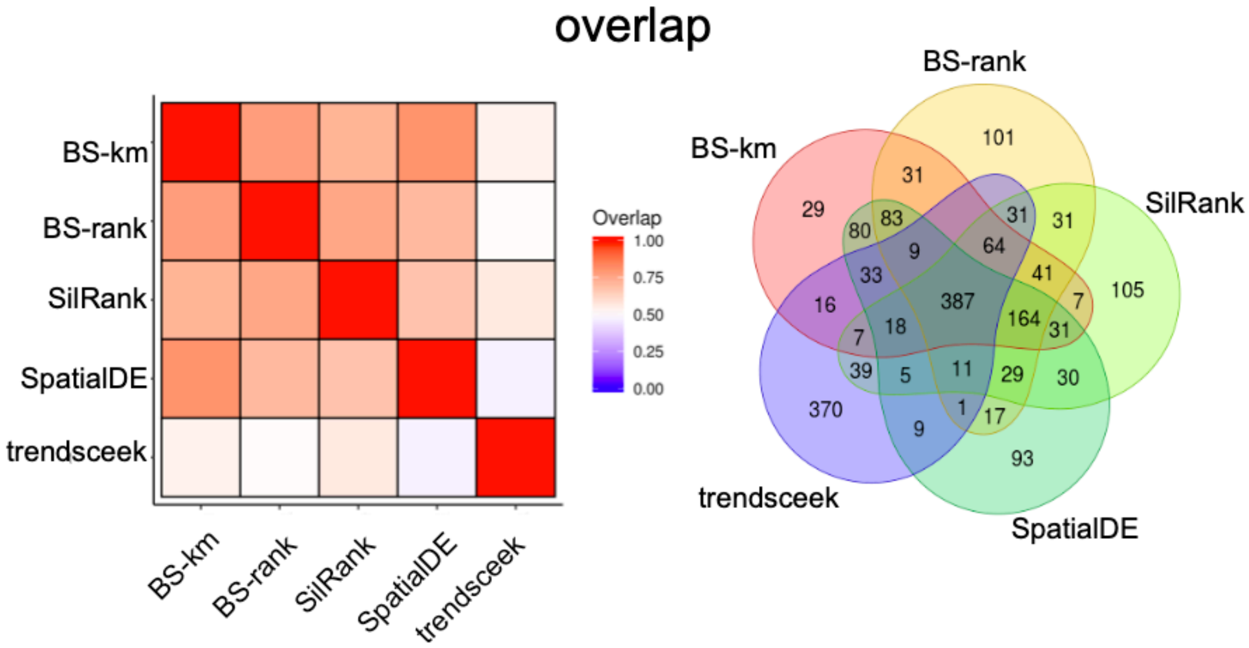
# Spatial analyses: spatial genes

bin_genes = binSpect()

sil_genes = silhouetteRank() [*1]

spatialDE_genes = spatialDE() [*2]

trend_genes = trendSceek() [*3]



# Spatial analyses: spatial genes

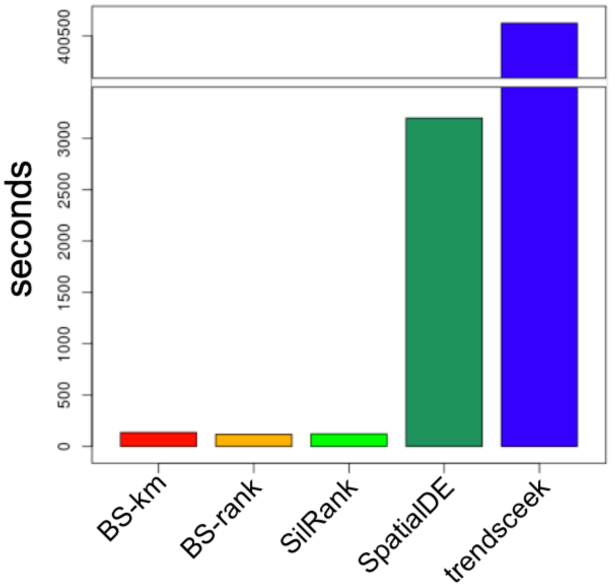
bin_genes = binSpect()

sil_genes = silhouetteRank() [*1]

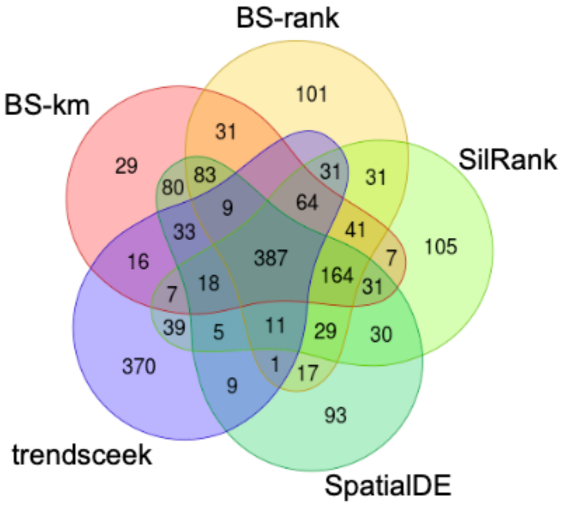
spatialDE_genes = spatialDE() [*2]

trend_genes = trendSceek() [*3]

speed



overlap

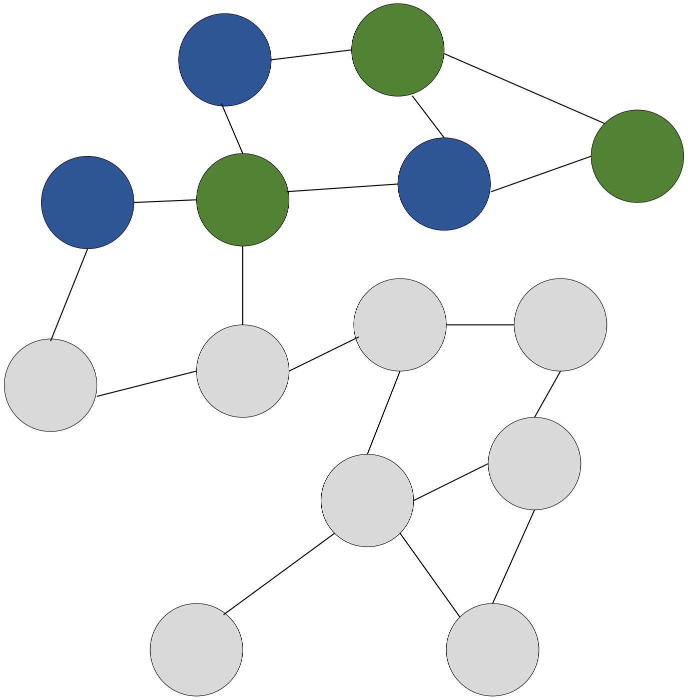


# Spatial analyses: spatial network

spatially co-expression modules

🔵 gene X **high**

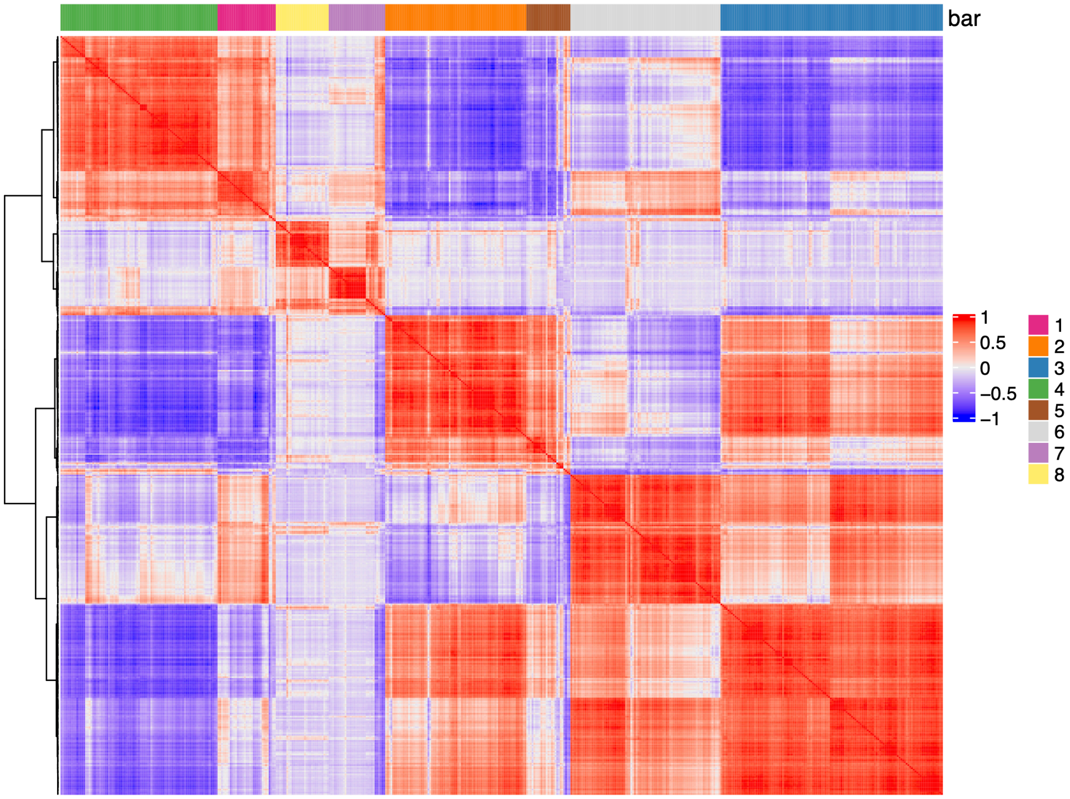
🟢 gene Y **high**



# Spatial analyses: co-expression modules

```
coexpr_modules = detectSpatialCorGenes ()
```

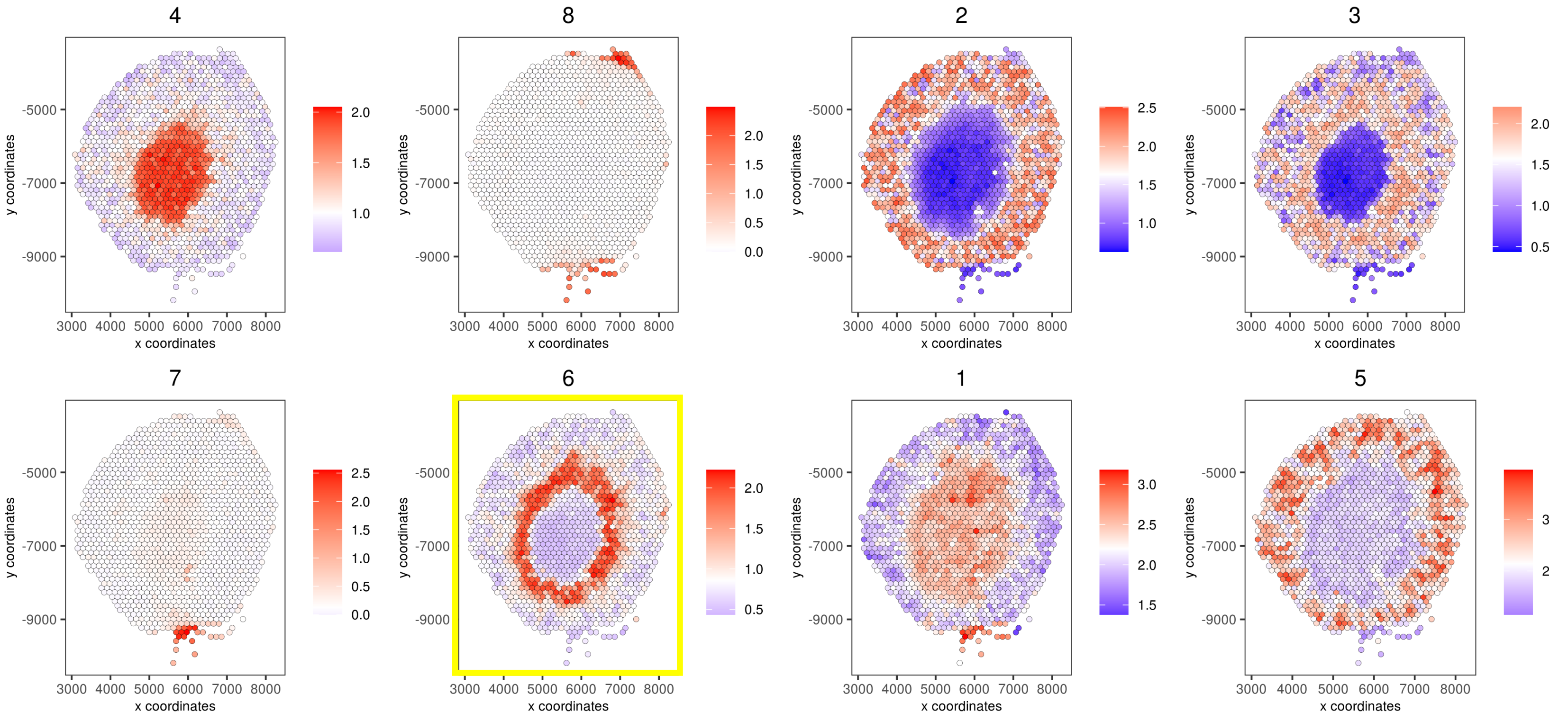
```
coexpr_modules =
clusterSpatialCorGenes (coexpr_modules)
```



# Spatial analyses: co-expression modules

```
coexpr_modules =
detectSpatialCorGenes ()
```

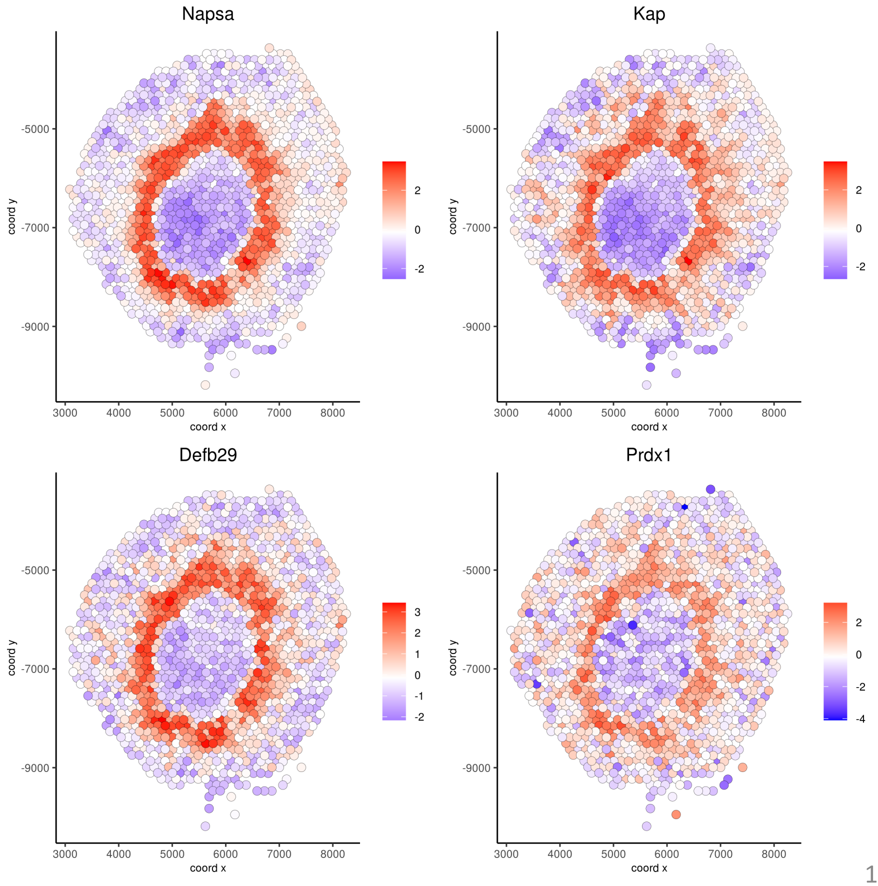
```
coexpr_modules =
clusterSpatialCorGenes (coexpr_modules)
```



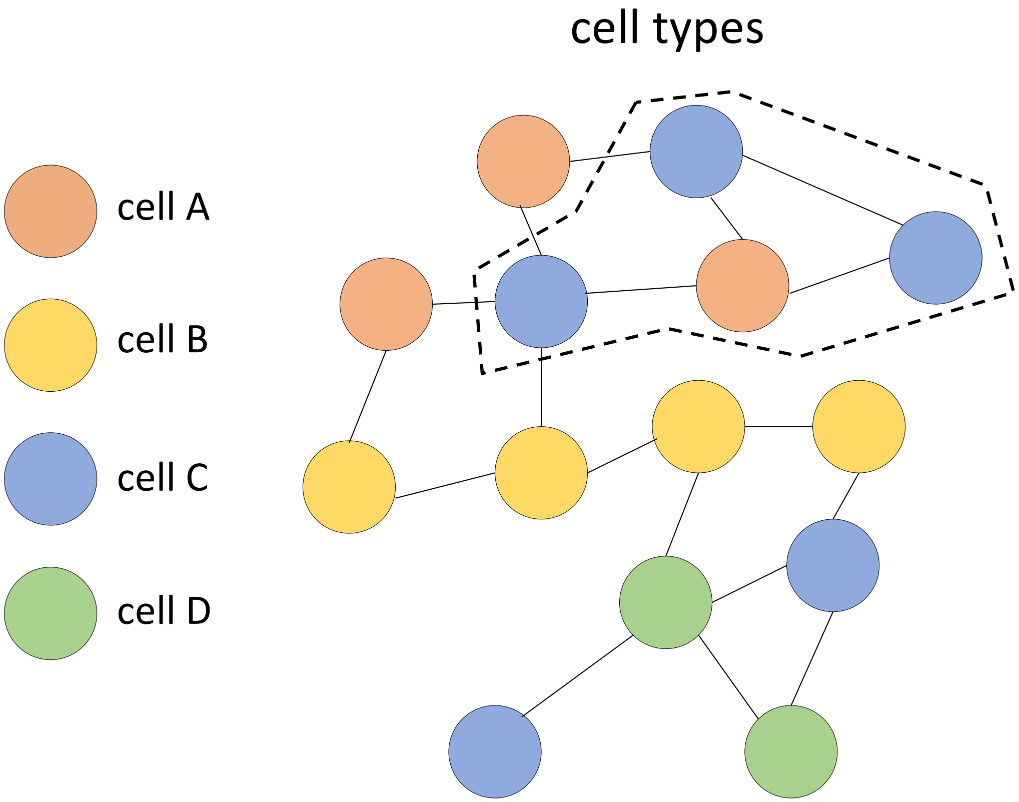
# Spatial analyses: co-expression modules

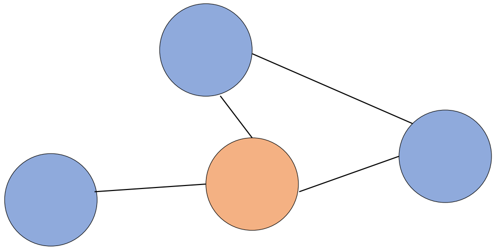
```
coexpr_modules = detectSpatialCorGenes
()
```

```
coexpr_modules =
clusterSpatialCorGenes (coexpr_modules)
```



# Spatial analyses: spatial network

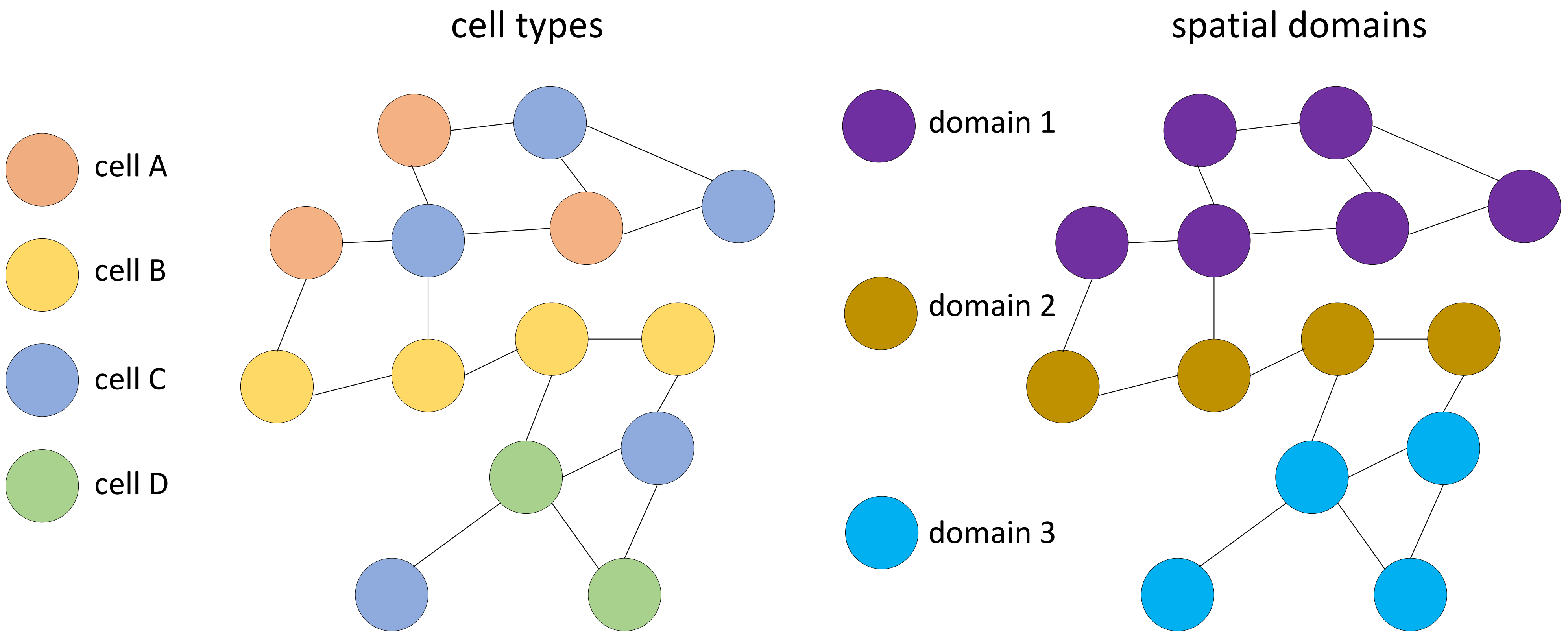




use HMRF to create discrete domains based on shared spatial gene expression patterns across cell types

Zhu, Q., et al. Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence *in situ* hybridization data. *Nature Biotechnology* (2018)

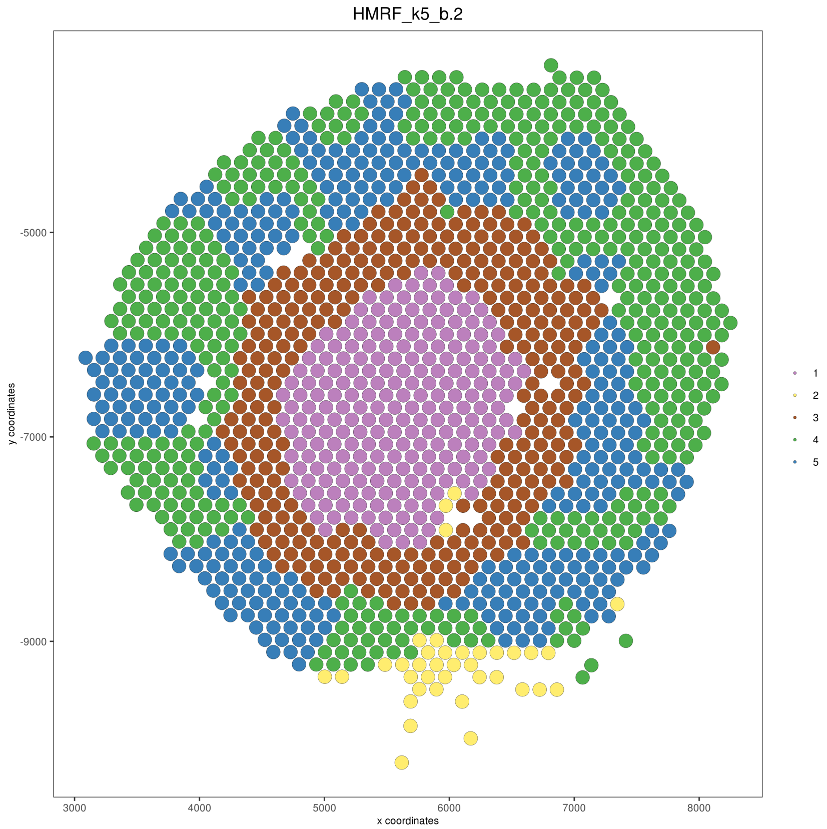
# Spatial analyses: spatial network



Zhu, Q., et al. Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence *in situ* hybridization data. *Nature Biotechnology* (2018)

# Spatial analyses: discrete domains

```
HMFR_results = doHMRF()
```



# Spatial analyses: cell-cell interaction

1. Spatial patterns:
    1. individual genes / proteins
    2. continuous spatial expression modules
    3. discrete spatial domains

2. Cell-cell interaction:
    1. neighborhood composition analysis
    2. Cell interaction induced gene expression changes
    3. spatial ligand-receptor activity

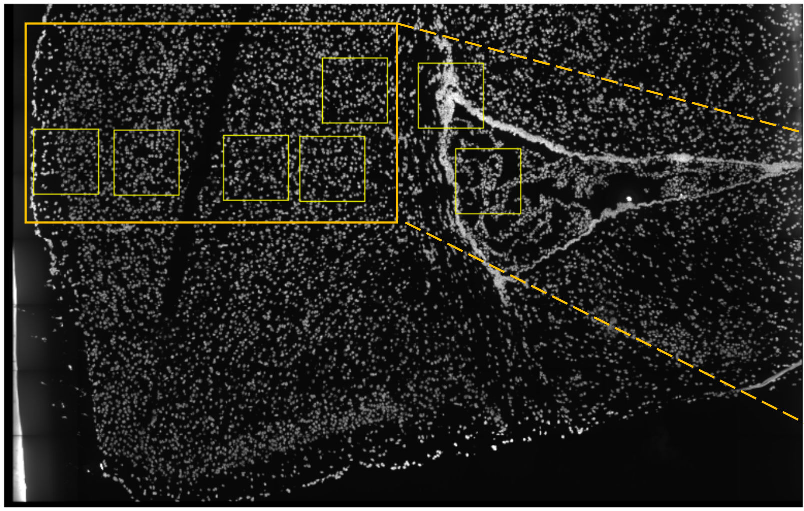
# Spatial analyses: cell-cell interaction with seqFISH+

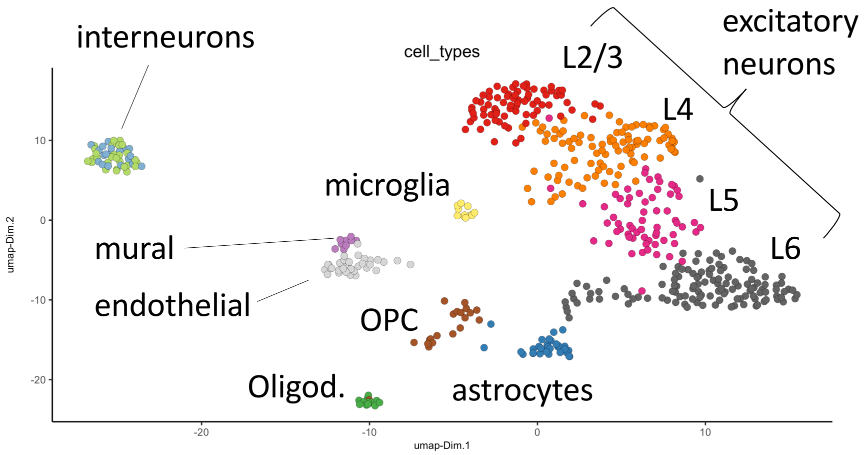
**Mouse SS cortex & subventricular zone**

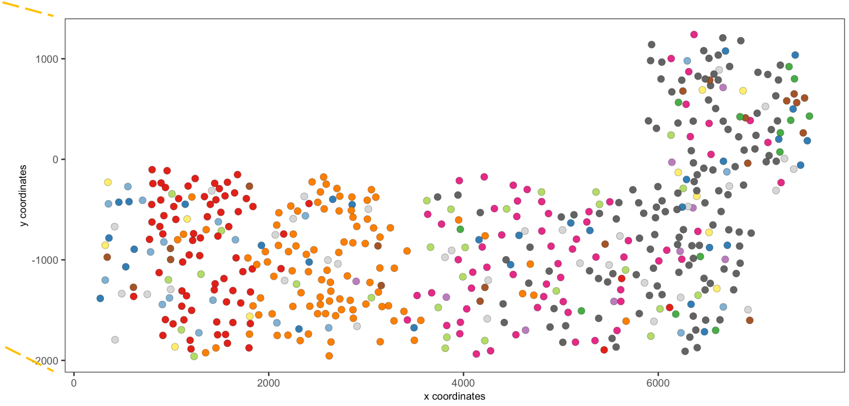
7 fields

913 cells

10.000 genes

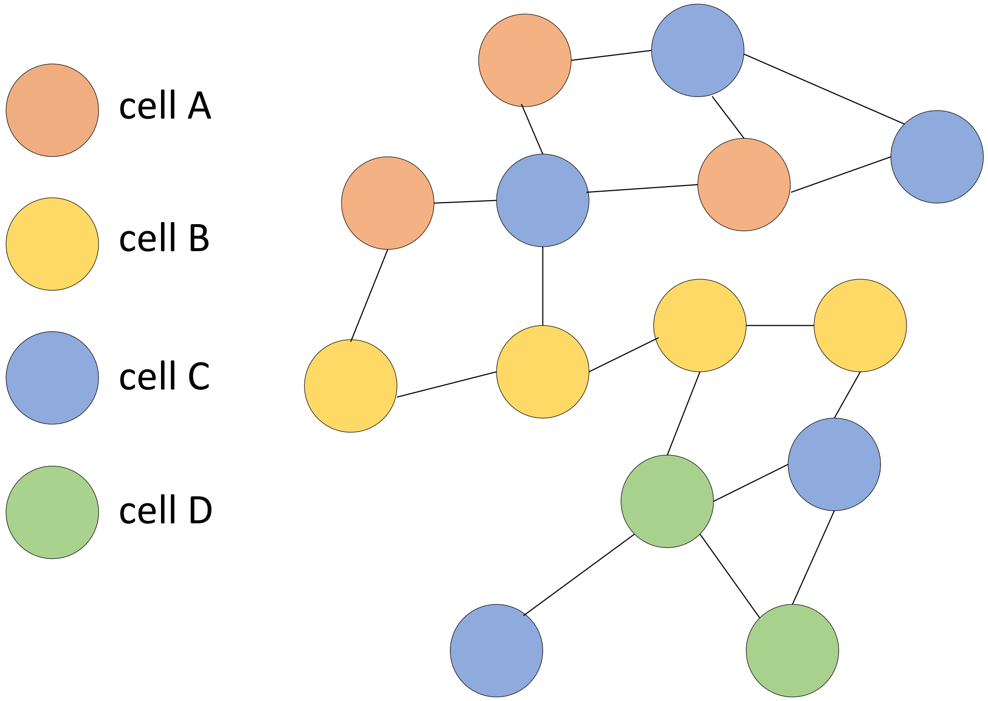






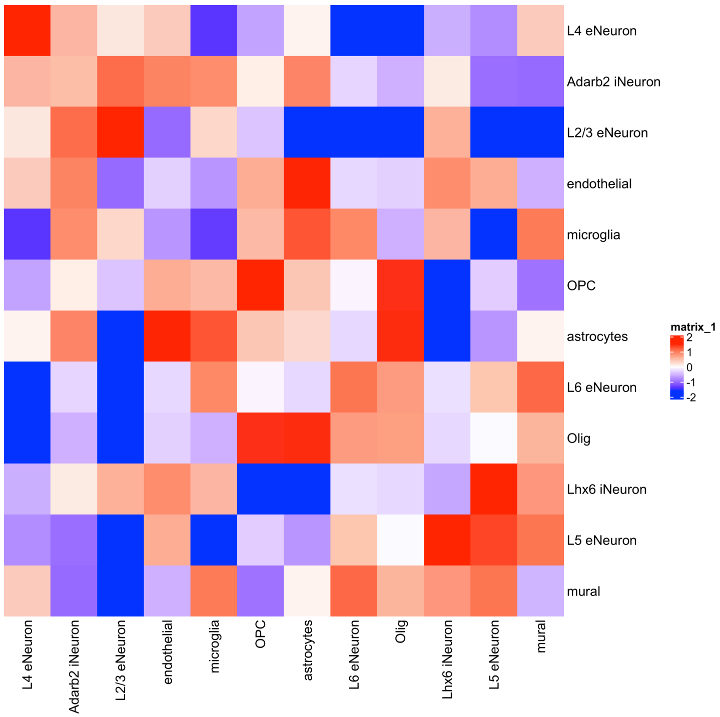
# Spatial analyses: neighborhood composition

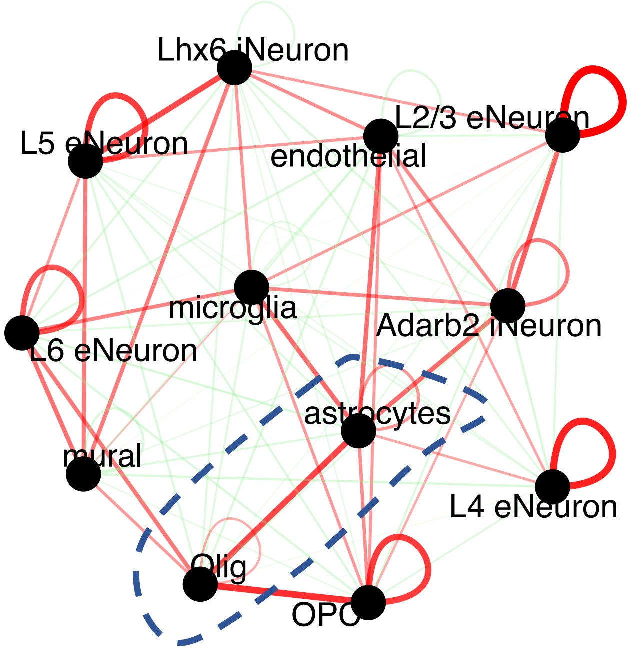
cell types



# Spatial analyses: neighborhood composition

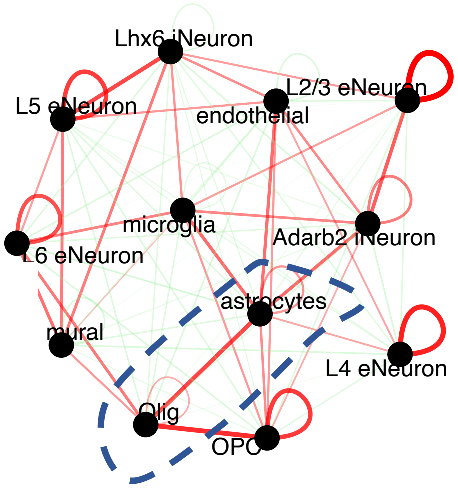
```
proximity_results = cellProximityEnrichment()
```

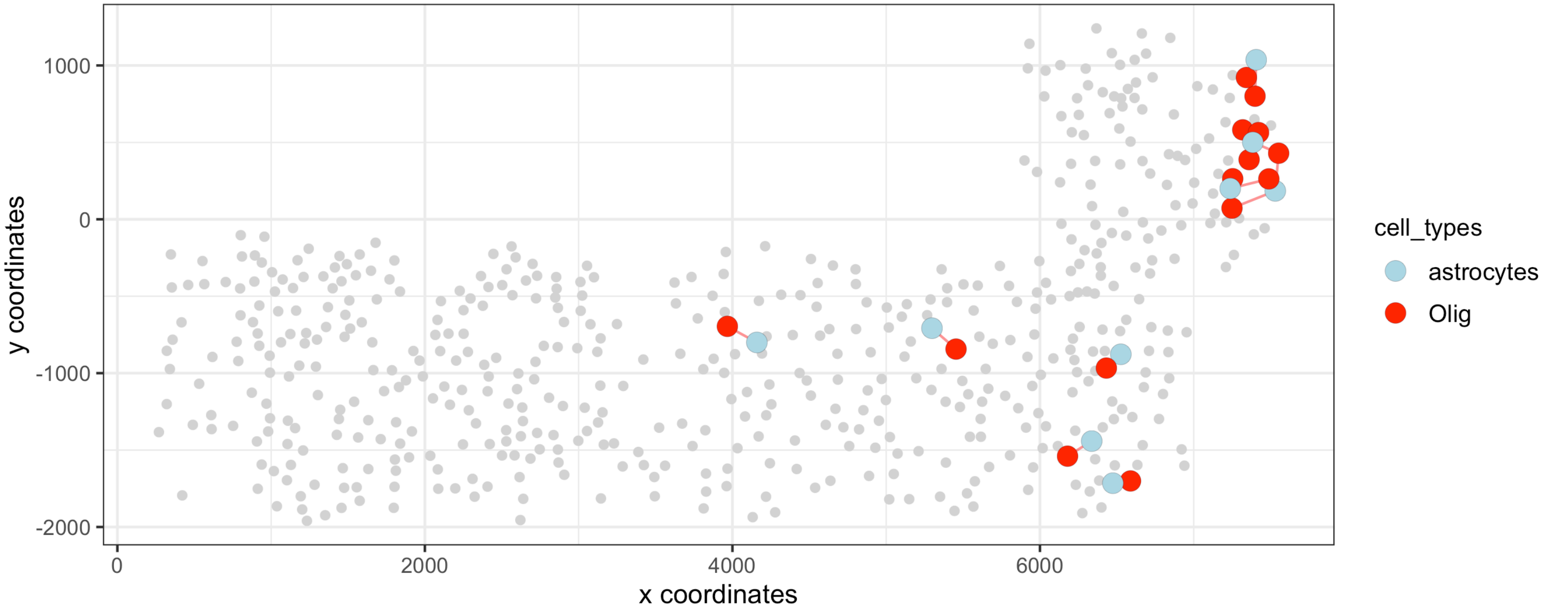




# Spatial analyses: neighborhood composition

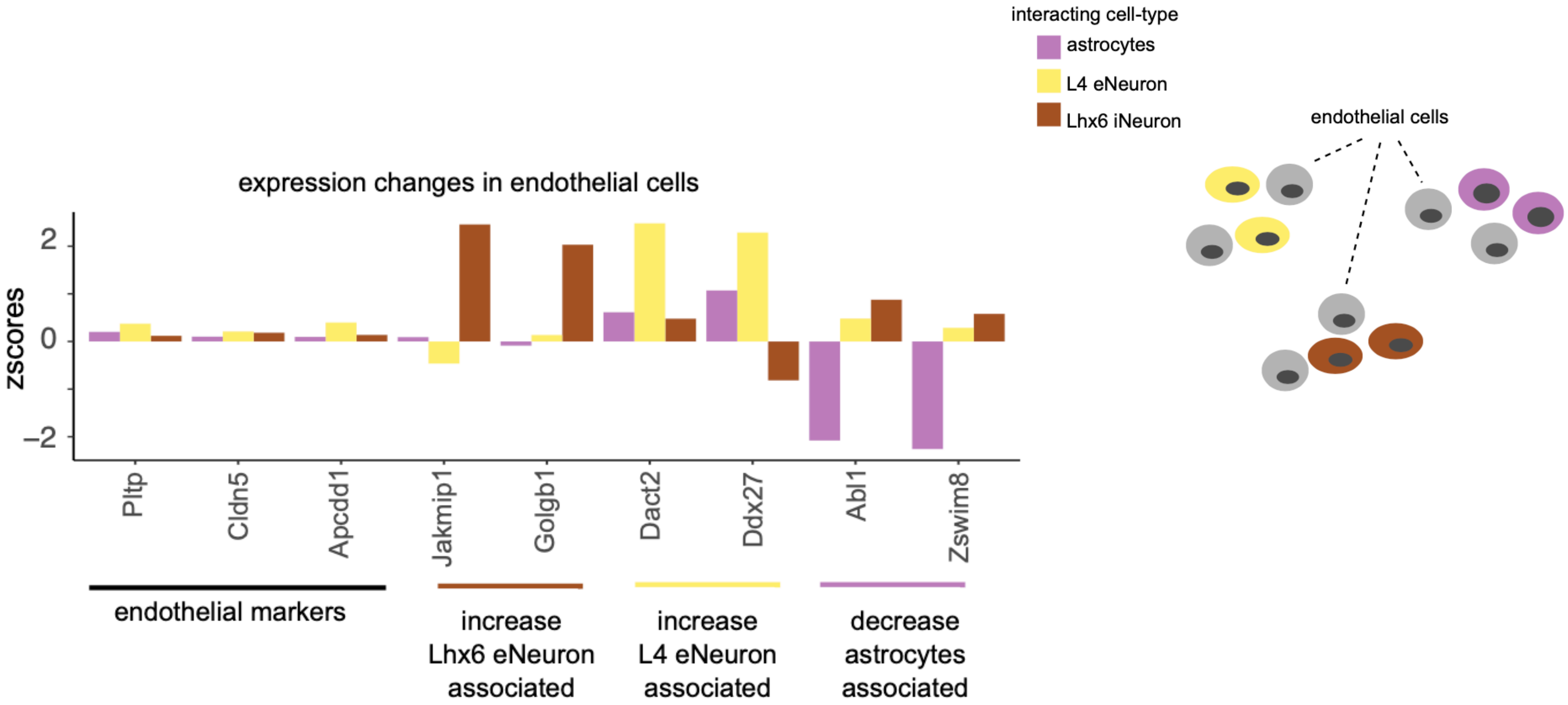
```
proximity_results = cellProximityEnrichment()
```





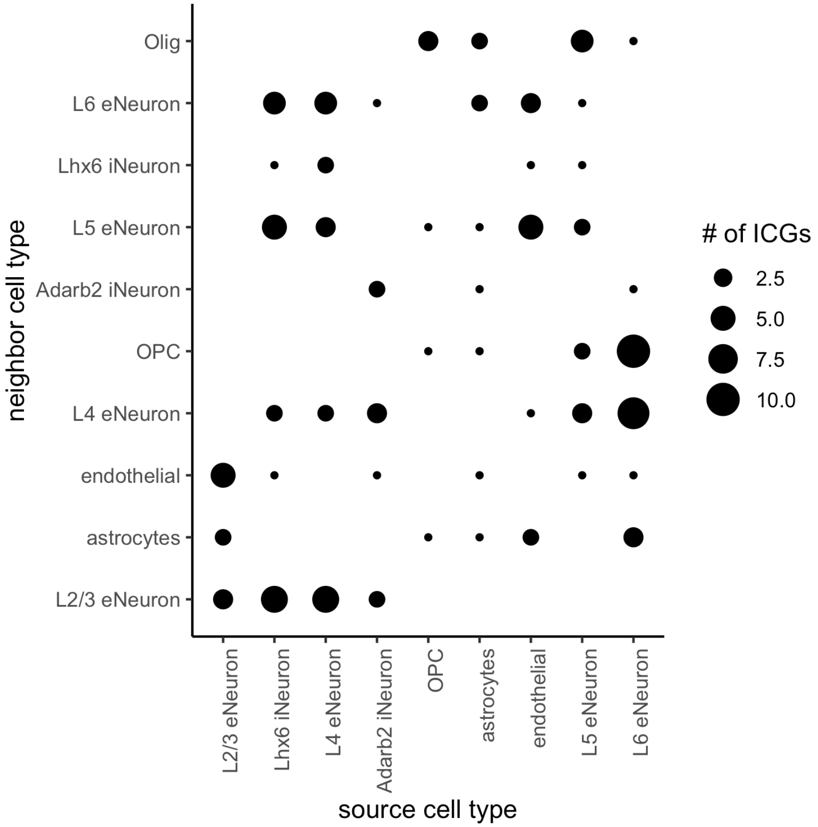
# Spatial analyses: interaction changed genes (ICG)

```
ICG_results = findCellProximityGenes()
```

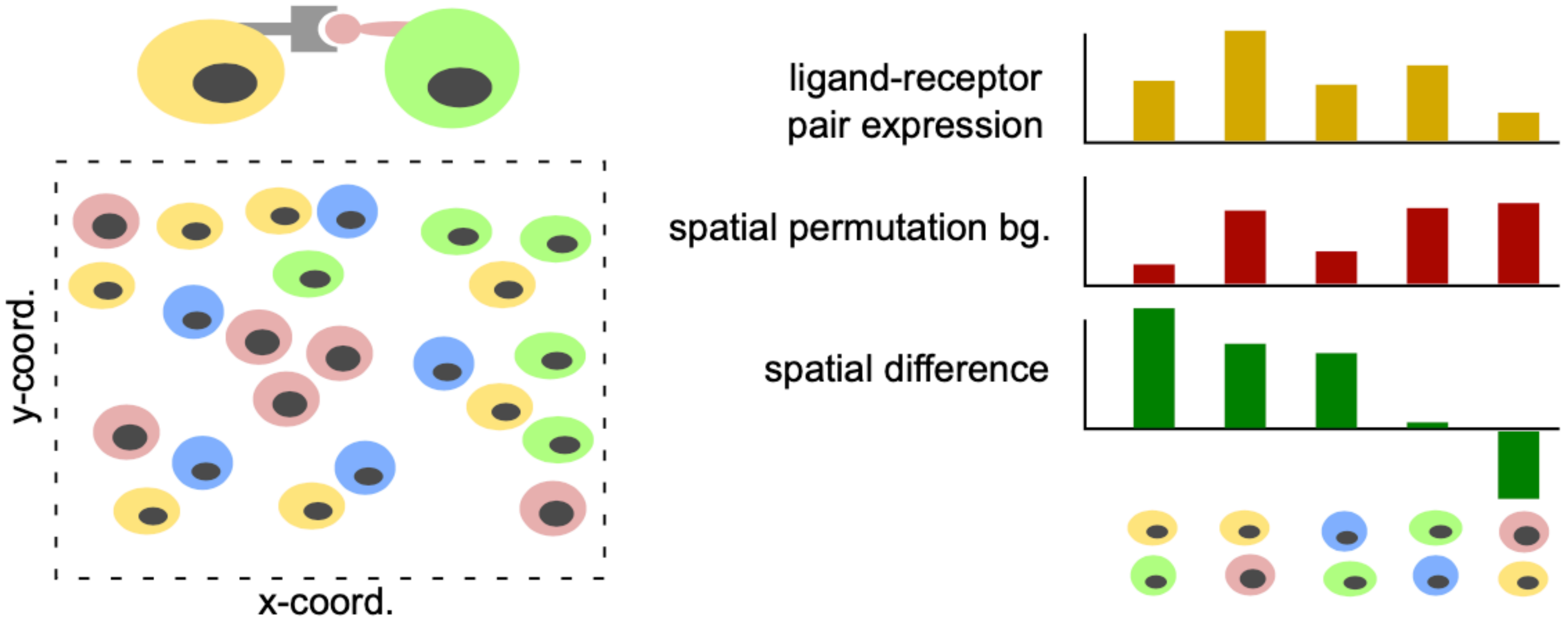


# Spatial analyses: interaction changed genes (ICG)

```
ICG_results = findCellProximityGenes()
```

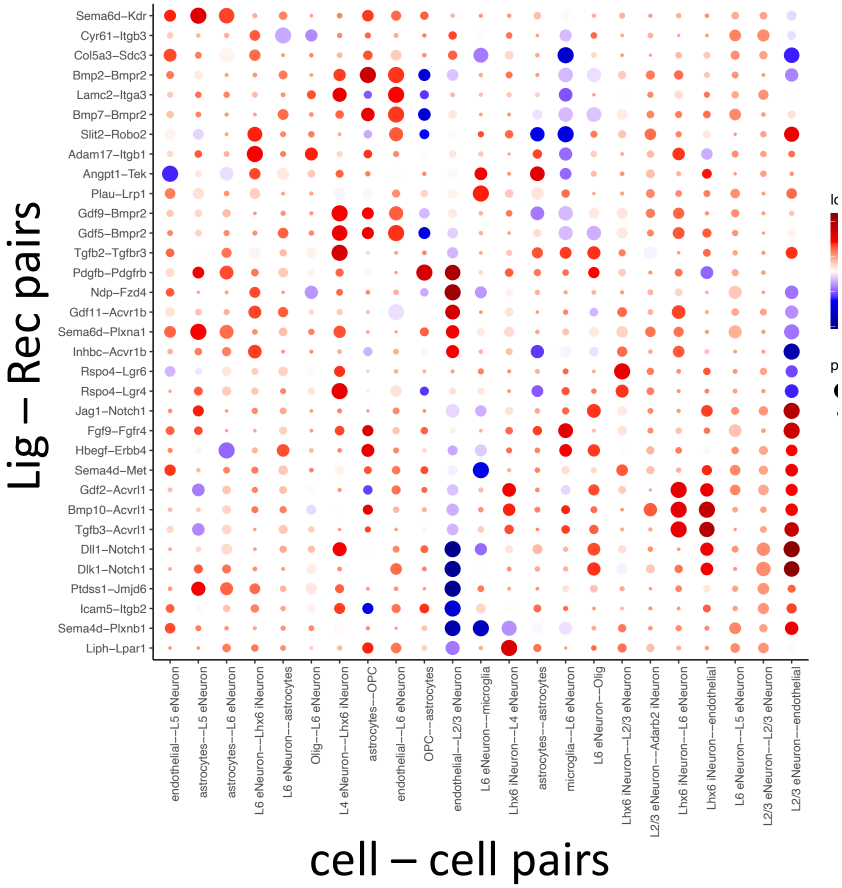


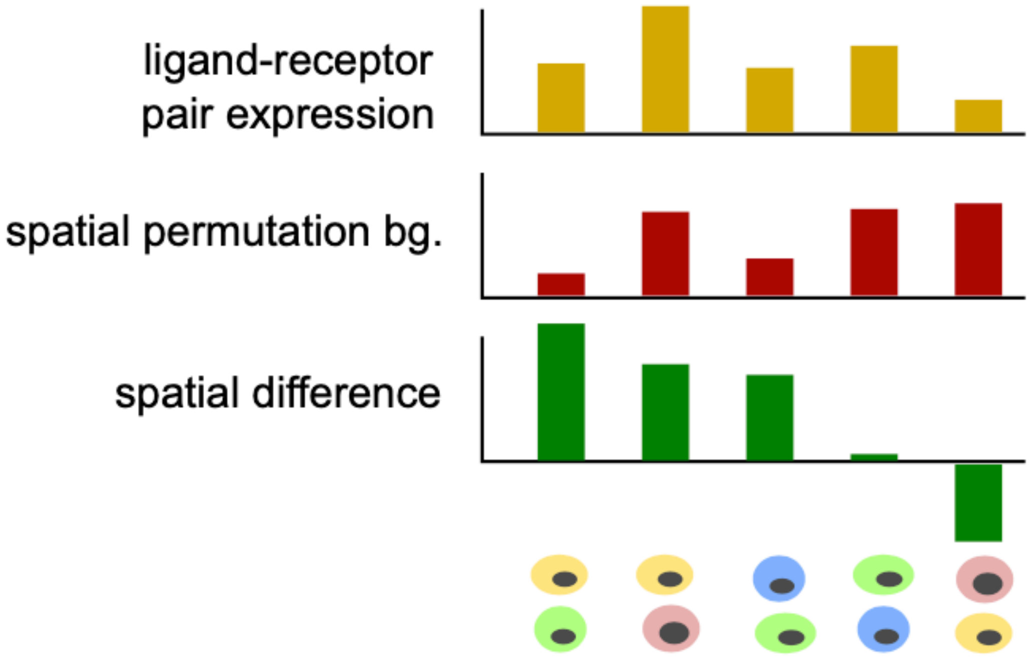
# Spatial analyses: spatial ligand receptor activity



# Spatial analyses: spatial ligand receptor activity

```
spatial_LR_res = spatCellCellcom()
```





ligand-receptor pair expression

spatial permutation bg.

spatial difference

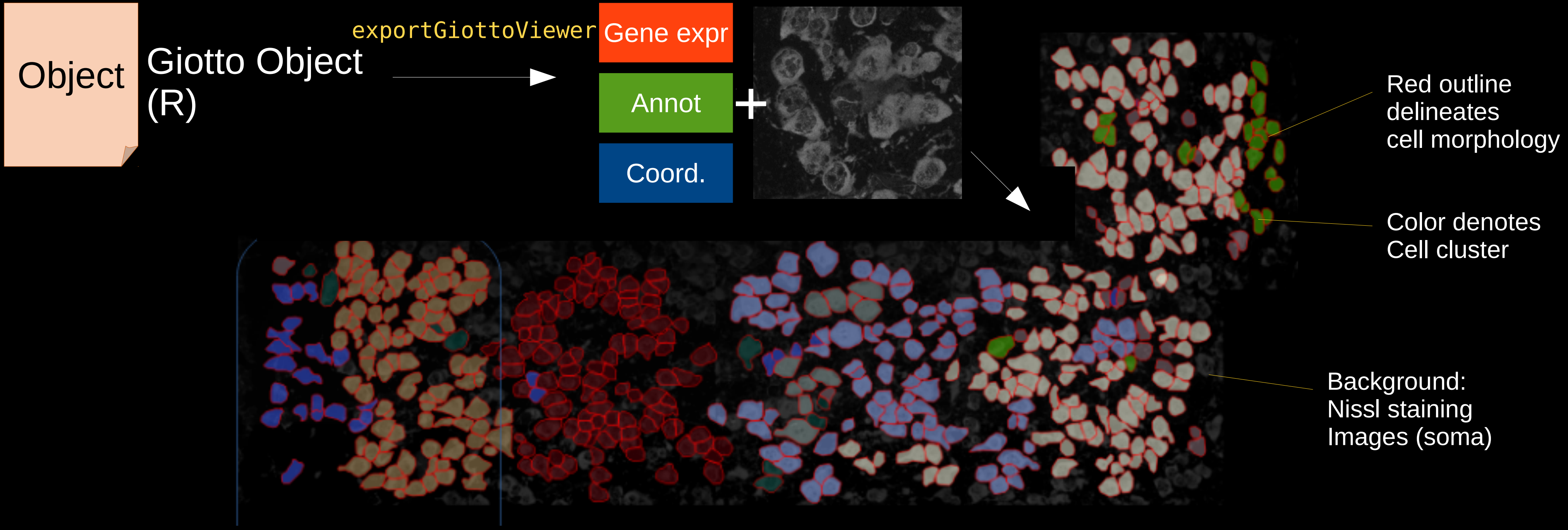
# Giotto Viewer – Interactive visualization of spatial gene expression data

Qian Zhu

@qzhu2012

# Giotto viewer – interactive visualization of spatial expression data

A locally hosted web site for displaying fluorescent images, cells, gene expression, transcripts, cellular morphologies in an interactive way.



Object

Giotto Object (R)

`exportGiottoViewer`

Gene expr

Annot

Coord.

+

Red outline delineates cell morphology

Color denotes Cell cluster

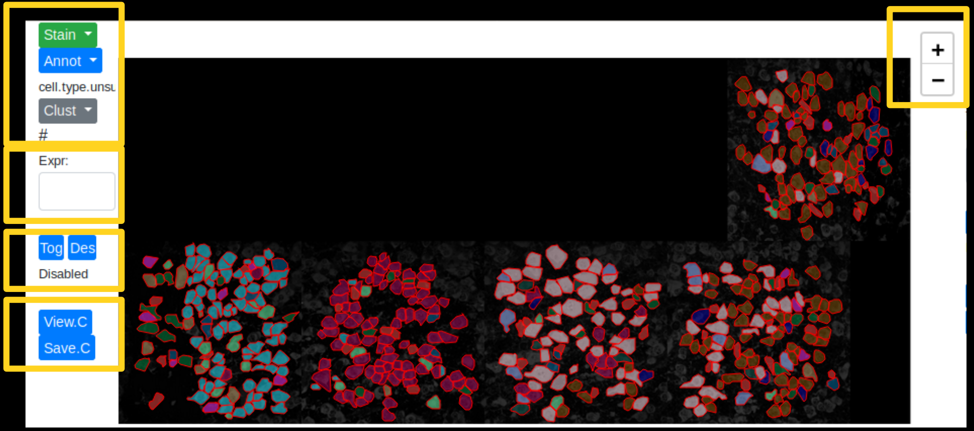
Background: Nissl staining Images (soma)

# Organization of Giotto viewer

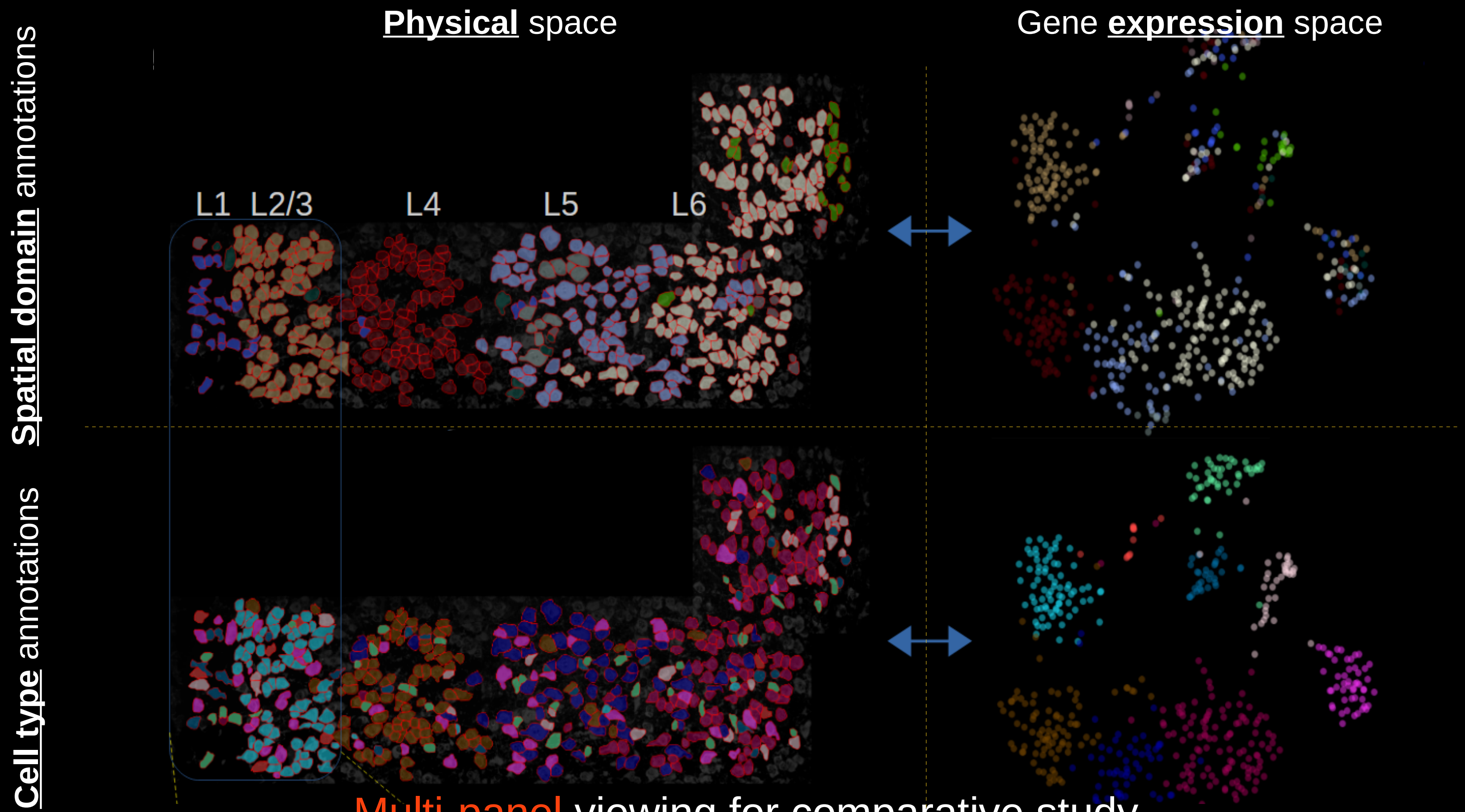
Interactive viewer highlights:

- Zooming, panning of canvas
- Customizable panel, users may choose 1 or more of:
  - Gene expression
  - Morphologies
  - Cell cluster-based coloring
  - Staining image
- Easy selection of cells
- Export of cell selection
- Search gene expression

- Support for transcript localization visualization



# Comparative feature of Giotto viewer



**Physical** space
Gene **expression** space
**Spatial domain** annotations
**Cell type** annotations
L1
L2/3
L4
L5
L6

Multi-panel viewing for comparative study

Interactivity between panels: linking, syncing

# Giotto viewer – 10X Visium dataset

10X Genomics Visium dataset

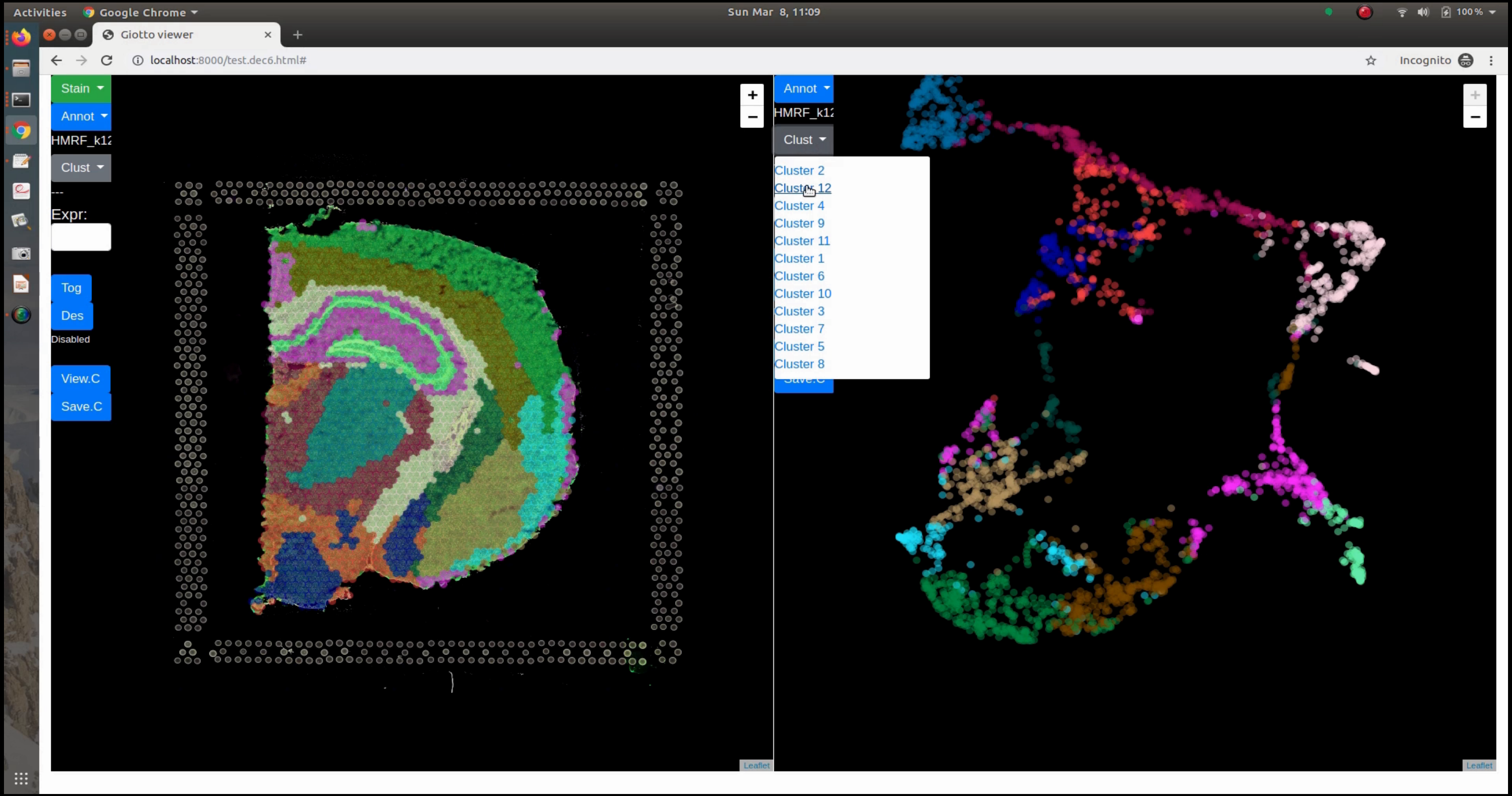
https://youtu.be/59l0g95OoU8

Mouse adult brain coronal slice

# Giotto viewer – 10X Visium dataset

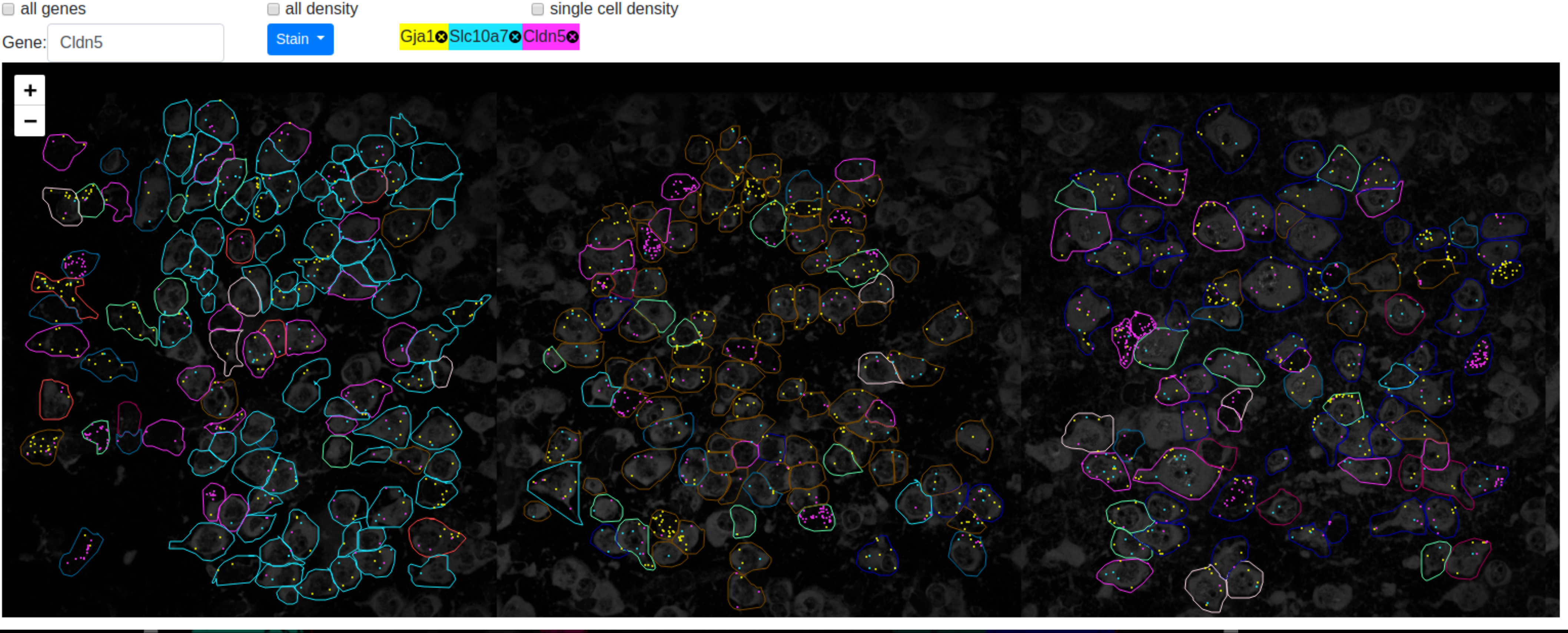
10X Genomics Visium dataset

https://youtu.be/59l0g95OoU8



Spots visualization
Cell morphologies
under spots

# Transcript visualization feature of Giotto Viewer



# Giotto viewer – interactive visualization of spatial expression data

Giotto viewer provides a good supplementary website for future spatial expression projects.

Interested in setting up?

Visit tutorial:

Giotto viewer: setting up a dataset.
http://spatialgiotto.rc.fas.harvard.edu/giotto.viewer.setup2.html

Giotto viewer: usage manual.
http://spatialgiotto.rc.fas.harvard.edu/documentation.html

@qzhu2012

# Giotto viewer set up: step1 JSON (preprocessing)

```
~/.local/bin/giotto_setup_image --require-stitch=y --image=y --image-multi-channel=y --
segmentation=y --multi-fov=y –output-json=step1.json

~/.local/bin/giotto_step1_modify_json --input step1.json --add-image
"segmentation_staining_1_MMStack_Pos[POSITION].ome.tif" --output step1.json
~/.local/bin/giotto_step1_modify_json --input step1.json --change-positions 0 1 2 3 4 --output
step1.json
~/.local/bin/giotto_step1_modify_json --input step1.json --change-stain-ids 0 4 7 --output
step1.json
~/.local/bin/giotto_step1_modify_json --input step1.json --change-offset offset.txt --output
step1.json
~/.local/bin/giotto_step1_modify_json --input step1.json --change-segmentation
"RoiSet_Pos[POSITION]_real.zip" --output step1.json
```

Automatically create
Step1 JSON file

@qzhu2012

# Giotto viewer set up: step1 JSON (preprocessing)

A sample, no need to create. Auto-created.

```
{
    "tiff_width": 4028,
    "tiff_height": 4028,
    "positions": [0, 1, 2, 3, 4],
    "stain_ids": [0, 1, 2, 3, 4],
       "offset": "GENERIC_offset.txt",
    "new_task_1": {
        "task": "decouple_tiff",
        "priority": 1,
        "input": "GENERIC_[POSITION].tif",
        "output_prefix": "pos[POSITION]",
        "positions": [0, 1, 2, 3, 4]
    },
    "new_task_2": {
        "task": "extract_roi_zip",
        "priority": 2,
        "input": "GENERIC_Roi_Pos[POSITION]_real.zip",
        "output": "roi/roi.pos[POSITION].all.txt",
        "tmp": "/tmp/pos[POSITION]",
        "positions": [0, 1, 2, 3, 4]
    },
    "new_task_3": {
        "task": "stitch_image",
        "priority": 3,
        "input": "pos[POSITION].[STAINID].tif",
        "output": "pos[STAINID].joined.tif",
        "offset": "offset.txt",
        "positions": [0, 1, 2, 3, 4],
        "stain_ids": [0, 1, 2, 3, 4]
    },
    "new_task_4": {
        "task": "stitch_coord",
        "priority": 4,
        "input": "GENERIC_centroids.csv",
        "output": "cell.centroid.stitched.pos.all.cells.txt",
        "offset": "offset.txt",
        "positions": [0, 1, 2, 3, 4]
    },
"new_task_5": {
        "task": "stitch_segmentation_roi",
        "priority": 5,
        "input": "roi/roi.pos[POSITION].all.txt",
        "output": "roi.stitched.pos.all.cells.txt",
        "offset": "offset.txt",
        "positions": [0, 1, 2, 3, 4]
    },
    "new_task_6": {
        "task": "align_segmentation_and_cell_centroid",
        "priority": 6,
        "input_cell_centroid": "cell.centroid.stitched.pos.all.cells.txt",
        "input_segmentation": "roi.stitched.pos.all.cells.txt",
        "output": "segmentation.to.cell.centroid.map.txt"
    },
    "new_task_7": {
        "task": "tiling_image",
        "priority": 7,
        "input": "Pos.ch[STAINID].joined.tif",
        "output_dir": "tiles.[STAINID]",
        "zoom": 6,
        "stain_ids": [0, 1, 2, 3, 4]
    },
    "new_task_8": {
        "task": "prepare_gene_expression",
        "priority": 8,
        "input": "giotto_expression.csv",
        "output_dir": "all.genes",
        "csv_sep": ",",
        "csv_header": 0,
        "csv_index_col": 0,
        "num_genes_per_file": 100
    }
}
```

@qzhu2012

# Giotto viewer set up: Step2 JSON (Panel configuration)

```
#2-panel:
~/.local/bin/giotto_setup_viewer --num-panel=2 --input-preprocess-
json=step1.json --panel-1=PanelPhysical --panel-2=PanelTsne --output-
json=step2.json —input-annotation-list=annotation_list.txt

#OR 4-panel:
~/.local/bin/giotto_setup_viewer --num-panel=4 --input-preprocess-
json=step1.json --panel-1=PanelPhysical --panel-2=PanelTsne --panel-
3=PanelPhysical --panel-4=PanelTsne --output-json=step2.json --input-
annotation-list=annotation_list.txt
```

Automatically create
Step2 JSON file

@qzhu2012

# Giotto viewer set up: Step2 JSON (Panel configuration)

```
{
        "num_panel": 4,
        "annotation_set": {
                "num_annot": 2,
                "annot_1": {
                        "file":
"testviewer/cell_types_annot_information.txt",
                        "name": "cell.type.unsup"
                },
                "annot_2": {
                        "file":
"testviewer/kmeans_annot_information.txt",
                        "name": "kmeans"
                }
        },
        "map_1": {
                "type": "PanelPhysical",
                "maxBound": 4096,
                "id": 1,
                "annot": "cell.type.unsup",
                "tile": "nissl",
                "dir_polyA": "imapr26.4",
                "dir_nissl": "imapr26.0",
                "dir_dapi": "imapr26.7",
                "gene_map": "10k.genes/gene.map",
                "segmentation_map":
"segmentation.to.cell.centroid.map.txt",
                "segmentation":
"roi.stitched.pos.all.cells.txt",
                "dir_gene_expression": "10k.genes",
                "gene_list": "gene.list.10k",
                "map_height": "500px"
        },
        "map_2": {
                "type": "PanelTsne",
                "maxBound": 500,
                "id": 2,
                "file_tsne": "test.cell.type.unsupervised.id.txt",
                "annot": "cell.type.unsup",
                "map_height": "500px"
        },
        "map_3": {
                "type": "PanelPhysical",
                "maxBound": 4096,
                "id": 3,
                "annot": "cell.type.unsup",
                "tile": "nissl",
                "dir_polyA": "imapr26.4",
                "dir_nissl": "imapr26.0",
                "dir_dapi": "imapr26.7",
                "gene_map": "10k.genes/gene.map",
                "segmentation_map": "segmentation.to.cell.centroid.map.txt",
                "segmentation": "roi.stitched.pos.all.cells.txt",
                "dir_gene_expression": "10k.genes",
                "gene_list": "gene.list.10k",
                "map_height": "500px"
        },
        "map_4": {
                "type": "PanelTsne",
                "maxBound": 500,
                "id": 4,
                "file_tsne": "test.cell.type.unsupervised.id.txt",
                "annot": "cell.type.unsup",
                "map_height": "500px"
        },
        "interact_1": ["map_1", "map_2", "map_3", "map_4"],
        "sync_1": ["map_1", "map_3"],
        "sync_2": ["map_2", "map_4"]
}
```

A sample, no need to create. Auto-created.

@qzhu2012

# Giotto viewer set up: step 3 (last step)
# Setting up the rest and launch the viewer

```
#do the step2 actions, and automatically generate webpages
~/.local/bin/smfish_read_config -c step2.json -o test.dec6.js -p test.dec6.html -q
test.dec6.css

#copy extra js and css folders
~/.local/bin/giotto_copy_js_css --output .

python3 -m http.server
```

Open your browser, navigate to http://localhost:8000/ and click on test.dec6.html.

@qzhu2012

# 10X Visium: Giotto viewer set up

Quick and easy instructions for 10X Visium dataset:

**Connecting with Giotto R package:**

```
#inside R, after finishing an analysis

# select annotations, reductions and expression values to view in Giotto Viewer
viewer_folder = paste0(results_folder, '/', 'mouse_visium_brain_viewer')
exportGiottoViewer(gobject = visium_brain,
                   output_directory = viewer_folder,
                   spat_enr_names = 'PAGE',
                   factor_annotations = c('in_tissue',
                                          'leiden_clus',
                                          'HMRF_k12_b.1'),
                   numeric_annotations = c('nr_genes',
                                           'Granule_neurons'),
                   dim_reductions = c('tsne', 'umap'),
                   dim_reduction_names = c('tsne', 'umap'),
                   expression_values = 'scaled',
                   expression_rounding = 2,
                   overwrite_dir = T):
```

If you have raw image, also put it in output_directory.

http://spatialgiotto.rc.fas.harvard.edu/giotto.visium.brain.html

@qzhu2012

# 10X visium: Giotto viewer set up

Quick and easy instructions
for 10X Visium dataset:

**In BASH:**

```
#create step1 json file
~/.local/bin/giotto_setup_image --require-stitch=n --image=y --image-multi-channel=n --segmentation=n --
multi-fov=n --output-json=step1.json
```

```
#automatically fill in image dimension in the step1 json file
~/.local/bin/giotto_step1_modify_json --add-image V1_Adult_Mouse_Brain_image.tif --input step1.json --
output step1.json
```

Needs to be raw image

```
#do the step1 actions
~/.local/bin/smfish_step1_setup -c step1.json
```

```
#create step2 json file
~/.local/bin/giotto_setup_viewer --num-panel=2 --input-preprocess-json=step1.json --panel-
1=PanelPhysical10X --panel-2=PanelTsne --output-json=step2.json --input-annotation-
list=annotation_list.txt
```

```
#do the step2 actions
~/.local/bin/smfish_read_config -c step2.json -o test.dec6.js -p test.dec6.html -q test.dec6.css
```

```
#copy extra js and css folders
~/.local/bin/giotto_copy_js_css --output .
```

```
python3 -m http.server
```

@qzhu2012

# Thank you for your attention.

@qzhu2012

Qian Zhu <zqian@jimmy.harvard.edu>