



Spatial Transcriptome with R programming

2023. 11. 13. Mon

Choi Lab

Korea University College of Medicine



Welcome to Spatial Transcriptomic session

<https://www.dropbox.com/scl/fo/sv3rreio33338vzv2cfhz/h?rlkey=p77rer9grfhli0g6l0s6mc4d2&dl=0>



Website 혹은 QR code를 이용해서 Dropbox에 있는 모든 파일들을 한 번에 다운받으시길 권장합니다.

Table of Contents

1. Introduction of dataset
2. [Visium] RCTD: Decomposition analysis
3. [Visium] Cellchat: Cell-cell interaction analysis
4. [Xenium] Seurat & Xenium Explorer: Visualization of ligand and receptor genes
5. Summary

1

Introduction

R object

| Steps | File name | Input/Output |
|----------|-------------------------------------|---|
| RCTD | Bioinfo2023_Breast_visium.rds | Visium pre-processed RCTD input |
| RCTD | Bioinfo2023_Breast_singlecell.rds | Single cell annotated RCTD input |
| RCTD | Bioinfo2023_Breast_RCTD.rds | RCTD output |
| RCTD | Bioinfo2023_Breast_visium_final.rds | Visium finally processed file CellChat input |
| Cellchat | Bioinfo2023_Breast_CellChat.rds | CellChat output |
| Xenium | Bioinfo2023_Breast_xenium.rds | Xenium processed file |

1. **Input files and python script** highlighted with purple should be downloaded before starting analysis
2. Please make a **directory** for this analysis and save necessary objects at the directory
3. Set the directory before we start analysis

Dataset introduction

a single breast cancer FFPE tissue block was assayed with
a trio of complementary technologies (Chromium, Visium, Xenium)

| Dataset | Information |
|-------------|---|
| Single cell | Chromium 5' and 3' Gene Expression data were collected from dissociated tumor cells |
| Visium | A 5 µm FFPE tissue section |
| Xenium | A 5 µm FFPE tissue section |

High resolution mapping of the breast cancer tumor microenvironment using integrated single cell, spatial and in situ analysis of FFPE tissue
Amanda Janesick, Robert Shelansky, Andrew D. Gottscho, Florian Wagner, Morgane Rouault, Ghezal Beliakoff, Michelli Faria de Oliveira,
Andrew Kohlway, Jawad Abousoud, Carolyn A. Morrison, Tingsheng Yu Drennon, Seayar H. Mohabbat, Stephen R. Williams, 10x
Development Teams, Sarah E.B. Taylor bioRxiv 2022.10.06.510405

Heads up for the information

We omitted following steps but offers R script

| | |
|--|--------------------------------|
| Single-cell pre-processing and annotation of cells | singlecell_preprocess_script.R |
| Visium pre-processing | visium_preprocess_script.R |
| Cottrazm | visium_cottrazm_script.R |

We want to highlight

Our single-cell and Visium annotation are different from published paper

We annotated one Visium spot with one cell type for cell-cell interaction downstream analysis

2

RCTD

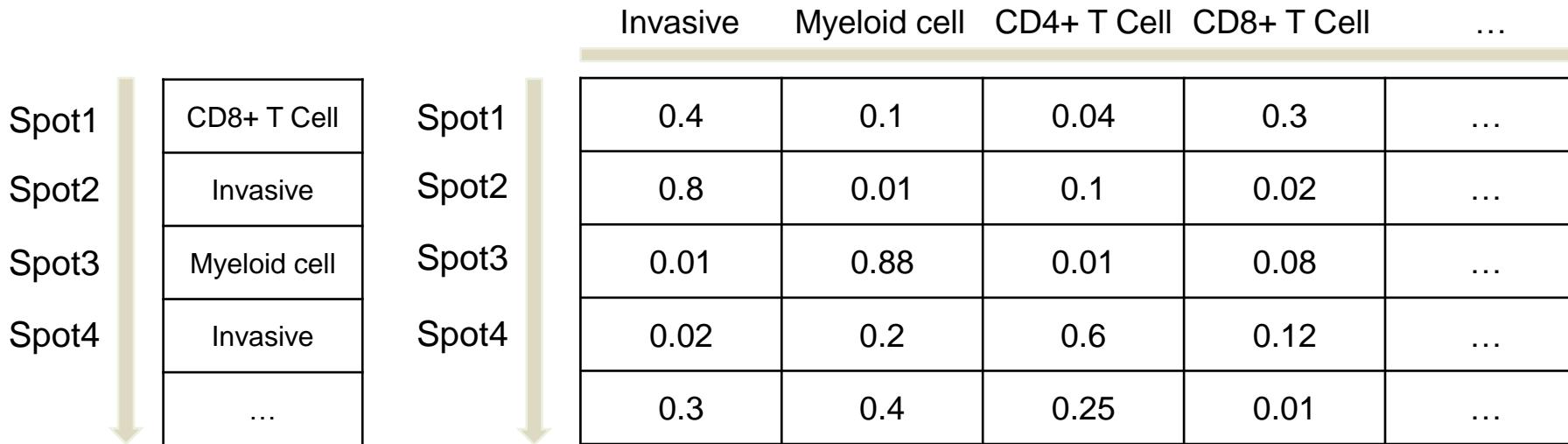
Spatial transcriptomics analysis

| Purpose and limitation | |
|-------------------------------|--|
| Purpose | Discovery of cell-type-specific spatial patterns of localization and expression. |
| Limitation | Individual measurements may contain contributions from multiple cells in Visium. |

Robust Cell Type Decomposition (RCTD) process

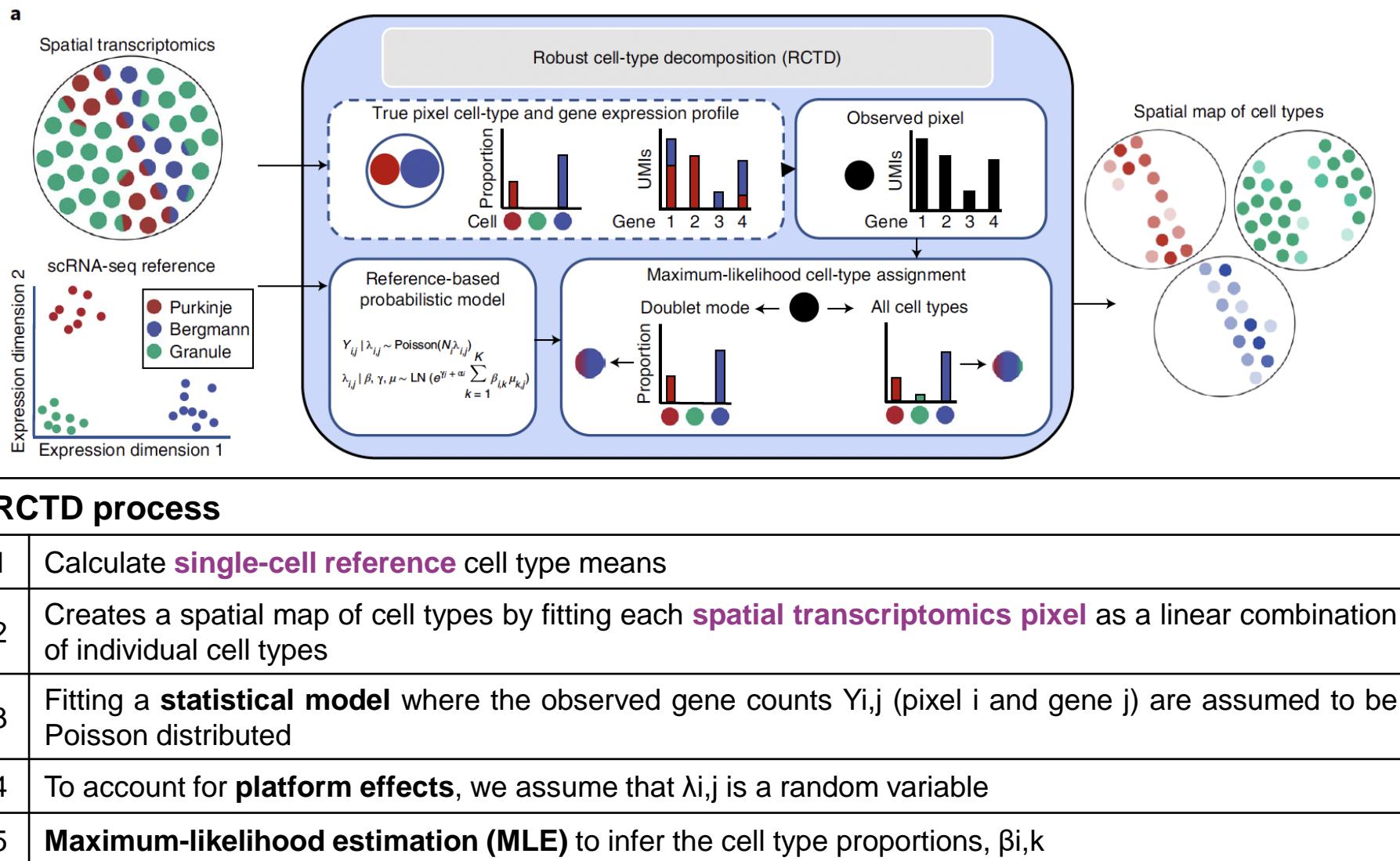


[1] Spatial mapping [2] Spatial deconvolution



| Analysis types | |
|-----------------------|---|
| Spatial Mapping | One representative cell type is assigned to individual spot. -> For cell-cell interaction downstream analysis |
| Spatial deconvolution | Proportions of multiple cell types is assigned to individual spot. |

RCTD algorithm



RCTD model

$$Y_{i,j} | \lambda_{i,j} \sim \text{Poisson} (N_i \lambda_{i,j})$$

$$\log (\lambda_{i,j}) = \alpha_i + \log \left(\sum_{k=1}^K \beta_{i,k} \mu_{k,j} \right) + \gamma_j + \varepsilon_{i,j}$$

Model counts with hierarchical model (Pixel I, Cell type K, Gene j)

| | |
|---------------------|---|
| $Y_{i,j}$ | The observed gene expression counts |
| $\lambda_{i,j}$ | Random variable to account for platform effects |
| $\mu_{k,j}$ | The mean gene expression profile for cell type k |
| γ_j | A gene-specific platform random effect |
| $\varepsilon_{i,j}$ | A random effect to account for gene-specific overdispersion . |
| α_i | A fixed pixel-specific effect |
| Goal | Estimate the $\beta_{i,k}$'s, which represent the cell type or cell types present in each pixel i |

Load RCTD input reference dataset (single cell)

Dropbox를 다운 받은 후에, 해당 폴더가 있는 경로로 설정합니다.

```
setwd("~/Downloads/BIOINFO2023") # MAC  
setwd("/Users/LG/Downloads/BIOINFO203") # Windows  
  
Breast_sc <- readRDS("./R_object/Bioinfo2023_Breast_singlecell.rds")
```

Prepare single cell dataset for RCTD input

```
counts_sc = Breast_sc$RNA@counts  
  
annotation_sc = Breast_sc$annotation  
names(annotation_sc) = rownames(Breast_sc@meta.data)  
annotation_sc = as.factor(annotation_sc)
```

Features {

Spots



```
> counts_sc[1:4,1:4]  
4 x 4 sparse Matrix of class "dgCMatrix"  
 1_AAACCCAGTGGAACCA-1 1_AACGCTCATATAGCC-1 1_AAACGCTGTTAGGCTT-1 1_AAAGGGCGTAAGAACT-1  
AL627309.1 . . . .  
AL627309.3 . . . .  
AL627309.5 . . . .  
AL627309.4 . . . .  
  
> annotation_sc[1:5]  
1_AAACCCAGTGGAACCA-1 1_AACGCTCATATAGCC-1 1_AAACGCTGTTAGGCTT-1 1_AAAGGGCGTAAGAACT-1 1_AAAGGGCGTAGTTCCA-1  
CD4+ T cell Invasive CD4+ T cell Invasive Invasive  
Levels: B cell CD4+ T cell CD8+ T cell DCIS #1 DCIS #2 DCIS #3 Invasive Mixed Myeloid cell Plasma cell Stromal
```

Prepare single cell dataset for RCTD input

```
nUMI_sc = Breast_sc@meta.data$nCount_RNA
names(nUMI_sc) = rownames(Breast_sc@meta.data)

reference = Reference(counts_sc, annotation_sc, nUMI_sc)
gc()

> nUMI_sc[1:5]
1_AAACCCAGTGGAACCA-1 1_AAACGCTCATATAGCC-1 1_AACCGCTGTTAGGCTT-1 1_AAAGGGCGTAAGAACT-1 1_AAAGGGCGTAGTTCCA-1
        4490           19092          1351          18918         38190
.
```

| reference | Large Reference (1 GB) |
|--------------------------|---|
| ..@ cell_types: | Factor w/ 13 levels "B cell", "CD4+ T cell", ... : 2 12 2 13 11 3 2 1 ... |
|- attr(*, "names")= | chr [1:20449] "1_AAACCCAGTGGAACCA-1" "1_AAACGCTCATATAGCC..." |
| ..@ counts : | Formal class 'dgCMatrix' [package "Matrix"] with 6 slots |
|@ i : | int [1:86281395] 9 41 128 140 151 153 201 254 258 259 ... |
|@ p : | int [1:20450] 0 1793 7035 7818 12438 18777 19470 20053 26127 ... |
|@ Dim : | int [1:2] 30962 20449 |
|@ Dimnames: | List of 2 |
|\$. : | chr [1:30962] "AL627309.1" "AL627309.3" "AL627309.5" "AL627309.4" ... |
|\$. : | chr [1:20449] "1_AAACCCAGTGGAACCA-1" "1_AAACGCTCATATAGCC-1" "1_AAA..." |
|@ x : | num [1:86281395] 1 1 1 2 1 1 1 1 1 1 ... |
|@ factors : | list() |
| ..@ nUMI : | Named num [1:20449] 4490 19092 1351 18918 38190 ... |
|- attr(*, "names")= | chr [1:20449] "1_AAACCCAGTGGAACCA-1" "1_AAACGCTCATATAGCC..." |

Process spatial dataset for RCTD input

```
breast_visium = readRDS("./R_object/Bioinfo2023_Breast_visium.rds")  
  
coords_visium = breast_visium@images$slice1@coordinates[,c("col","row")]  
  
counts_visium = breast_visium@assays$Spatial@counts
```

```
> head(breast_visium@images$slice1@coordinates)  
    tissue row col imagerow imagecol  
AACACCTACTATCGAA-1    1   0 122     4636     4131  
AACACGTGCATCGCAC-1    1  76  22    16640    13355  
AACACTTGGCAAGGAA-1    1  47  71    12067     8845  
AACAGGAAGAGCATAG-1   1  69   7    15518    14716  
AACAGGATTCATAGTT-1   1  49  43    12365    11404  
AACAGGCCAACGATTA-1   1  71 127    15920     3761  
  
> counts_visium[1:4,1:4]  
4 x 4 sparse Matrix of class "dgCMatrix"  
           AACACCTACTATCGAA-1 AACACGTGCATCGCAC-1 AACACTTGGCAAGGAA-1 AACAGGAAGAGCATAG-1  
SAMD11                 .                   2                   1                   .  
NOC2L                  .                   .                   3                   .  
KLHL17                 .                   .                   .                   .  
PLEKHN1                 .                   .                   1                   .
```

Process spatial dataset for RCTD input

```
nUMI_visium = colSums(counts_visium)

query = SpatialRNA(coords_visium, counts_visium, nUMI_visium)
gc()
```

```
> nUMI_visium[1:5]
AACACCTACTATCGAA-1 AACACGTGCATCGCAC-1 AACACTGGCAAGGAA-1 AACAGGAAGAGCATAG-1 AACAGGATTTCATAGTT-1
    12675           7886          32614          7484          6694
```

| query | Large SpatialRNA (359.4 MB) |
|---|---|
| ..@ coords:'data.frame': | 4992 obs. of 2 variables: |
|\$ x: int [1:4992] | 122 22 71 7 43 127 86 41 6 10 ... |
|\$ y: int [1:4992] | 0 76 47 69 49 71 28 51 24 12 ... |
| ..@ counts:Formal class 'dgCMatrix' [package "Matrix"] with 6 slots | |
|@ i : int [1:29737138] | 5 7 13 18 20 24 25 26 29 31 ... |
|@ p : int [1:4993] | 0 6022 10001 19018 23201 26894 29842 3... |
|@ Dim : int [1:2] | 18085 4992 |
|@ Dimnames:List of 2 | |
|\$. : chr [1:18085] | "SAMD11" "NOC2L" "KLHL17" "PLEKHN1" ... |
|\$. : chr [1:4992] | "AACACCTACTATCGAA-1" "AACACGTGCATCGCAC-1" ... |
|@ x : num [1:29737138] | 1 1 1 2 1 5 1 7 1 1 ... |
|@ factors : list() | |
| ..@ nUMI : Named num [1:4992] | 12675 7886 32614 7484 6694 ... |
|- attr(*, "names")= chr [1:4992] | "AACACCTACTATCGAA-1" "AACACGTGCAT..." |

Run RCTD in doublet mode

```
RCTD = create.RCTD(query, reference, max_cores = 8)
RCTD = run.RCTD(RCTD, doublet_mode = 'doublet')
# RCTD = readRDS("./Bioinfo2023_Breast_RCTD.rds")
RCTD_results = RCTD$results$results_df
breast_visium = AddMetaData(breast_visium, metadata = RCTD_results)
```

```
> head(RCTD$results$results_df)
```

| | spot_class | first_type | second_type | first_class | second_class | min_score | singlet_score | conv_all | conv_doublet |
|--------------------|-----------------|--------------|--------------|-------------|--------------|-----------|---------------|----------|--------------|
| AACACCTACTATCGAA-1 | doublet_certain | Stromal cell | Myeloid cell | FALSE | FALSE | 2462.049 | 2885.129 | TRUE | TRUE |
| AACACGTGCATCGCAC-1 | doublet_certain | Stromal cell | Myeloid cell | FALSE | FALSE | 2000.951 | 2441.102 | TRUE | TRUE |
| AACACTTGGCAAGGAA-1 | doublet_certain | Stromal cell | DCIS #1 | FALSE | FALSE | 4226.069 | 5101.309 | TRUE | TRUE |
| AACAGGAAGAGCATAG-1 | doublet_certain | Stromal cell | Myeloid cell | FALSE | FALSE | 2120.664 | 2622.367 | TRUE | TRUE |
| AACAGGGATTCTAGTT-1 | doublet_certain | Stromal cell | Plasma cell | FALSE | FALSE | 1899.106 | 2289.688 | TRUE | TRUE |
| AACAGGCCAACGATTA-1 | doublet_certain | Stromal cell | Myeloid cell | FALSE | FALSE | 1515.789 | 1806.427 | TRUE | TRUE |

| Analysis mode | |
|---------------|---|
| Doublet | Fits at most two cell types per pixel |
| Full | No restrictions on number of cell types, recommended for low spatial resolution technologies such as Visium |
| Multi | Finitely many cell types per pixel, e.g. 3 or 4. |

Process RCTD decomposed file

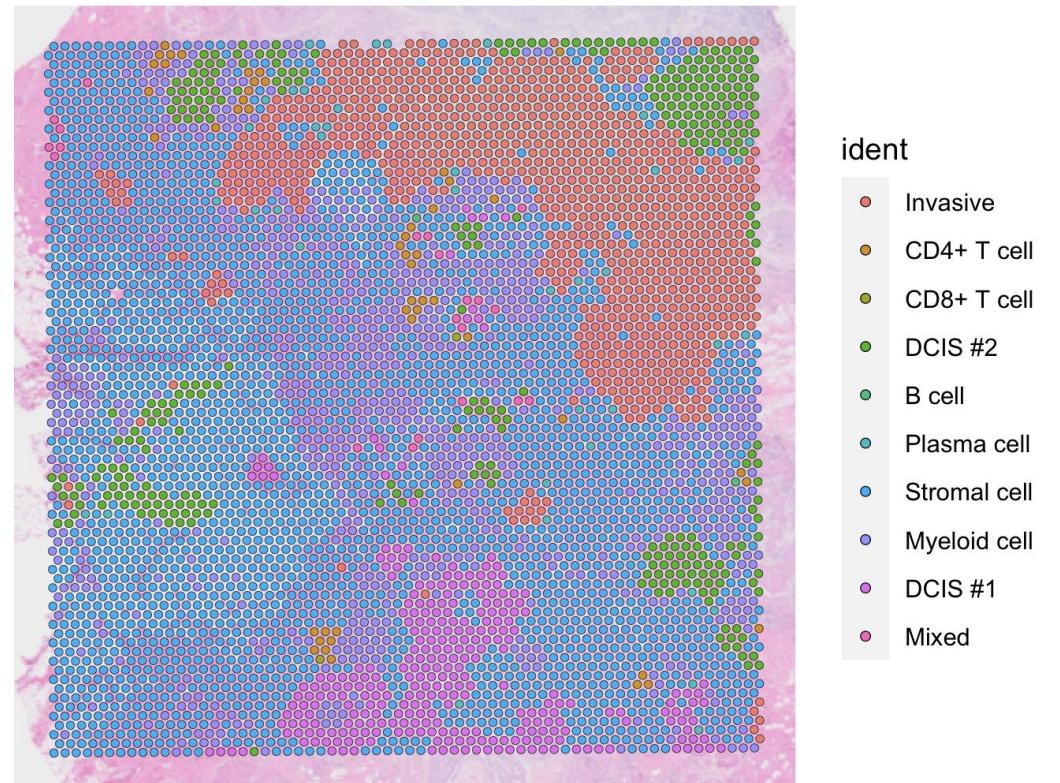
```
breast_visium = SetIdent(breast_visium, value="first_type")
table(breast_visium$first_type)
```

```
table(breast_visium$first_type)
```

| Invasive | CD4+ T cell | CD8+ T cell | DCIS #2 | B cell | Plasma cell | Stromal cell | Myeloid cell | DCIS #1 | Mixed |
|----------|-------------|-------------|---------|--------|-------------|--------------|--------------|---------|-------|
| 925 | 51 | 2 | 331 | 2 | 40 | 2571 | 745 | 300 | 21 |

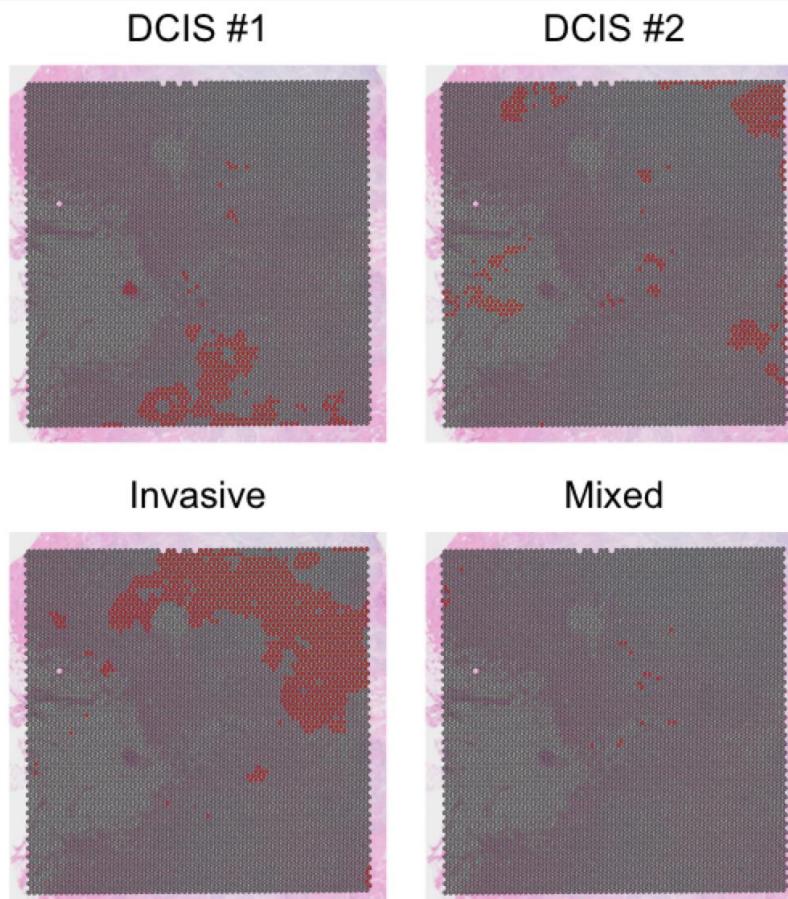
Breast spatial map of predicted cell type by RCTD

```
breast_visium <- subset(breast_visium, first_type %in%  
                           names(table(breast_visium$first_type)))  
breast_visium$first_type <- factor(breast_visium$first_type)  
SpatialDimPlot(breast_visium)
```



Spatial map of predicted cell type by RCTD (1)

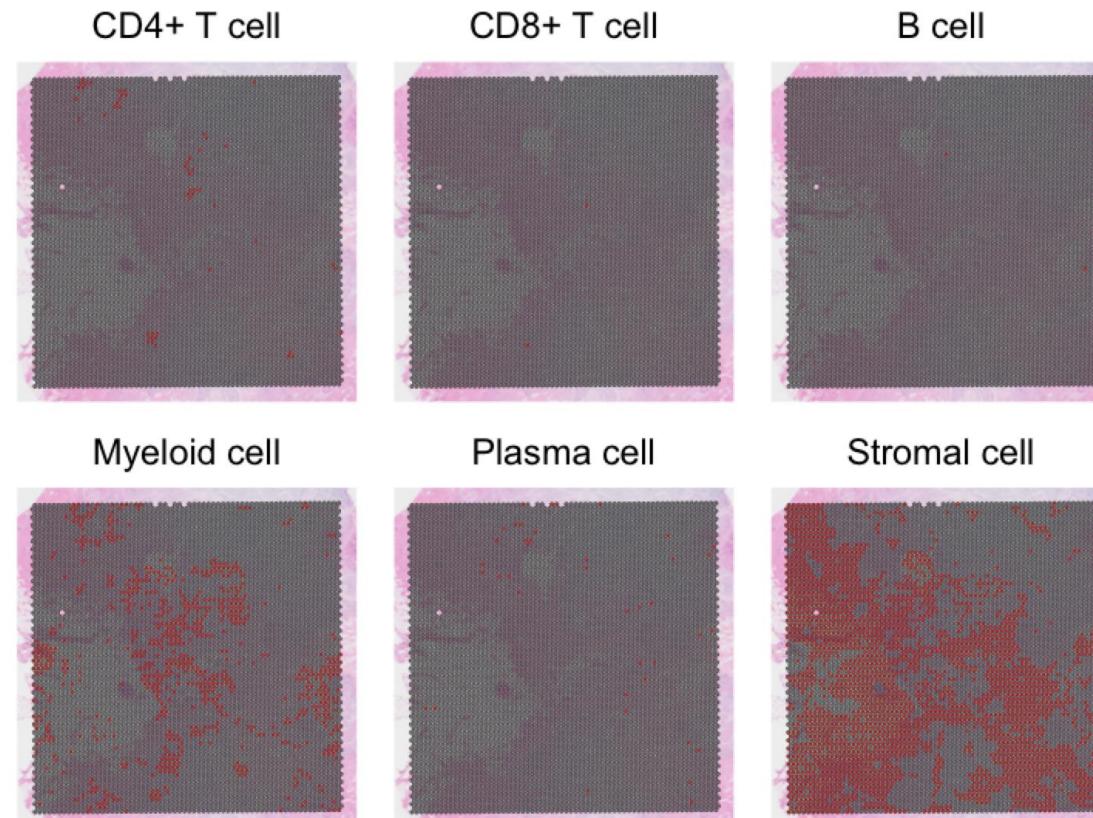
```
SpatialDimPlot(breast_visium, cells.highlight = CellsByIdentities(object = breast_visium,  
idents = c('DCIS #1','DCIS #2','Invasive','Mixed')), facet.highlight = TRUE)
```



| Cancer types | |
|--------------|---|
| DCIS | low-grade and high-grade ductal carcinoma in situ |
| Invasive | invasive carcinoma |

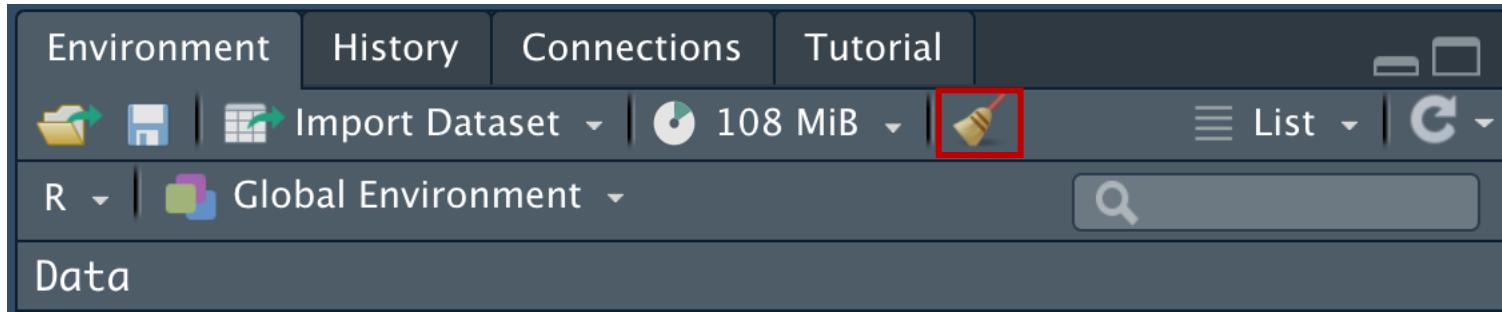
Spatial map of predicted cell type by RCTD (2)

```
SpatialDimPlot(breast_visium, cells.highlight = CellsByIdentities(object = breast_visium,  
    idents = c('CD4+ T cell', 'CD8+ T cell', 'B cell', 'Myeloid cell', 'Plasma cell', 'Stromal  
cell')), facet.highlight = TRUE, ncol = 3)
```



Remove all objects before starting next chapter

- Clear objects from the workspace.



```
# clean up memory in R  
gc()
```

3

Cell-cell interaction analysis - CellChat

What is CellChat?

CellChat is an useful tool to **quantitatively infer and analyze intercellular communication networks** from single-cell RNA-sequencing data and spatial transcriptomics data.

Requires **gene expression** and **spatial location data** of spots/cells as the user input and models the probability of cell-cell communication by integrating gene expression with spatial distance as well as prior knowledge of the interactions between signaling ligands, receptors and their cofactors.



JIN, Suoqin, et al. Inference and analysis of cell-cell communication using CellChat. Nature communications, 2021, 12.1: 1-20.
https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/CellChat_analysis_of_spatial_imaging_data.html

Load data

```
# Load cell type annotated visium data and visualization
visium.breast = readRDS("./R_object/Bioinfo2023_Breast_visium_final.rds")

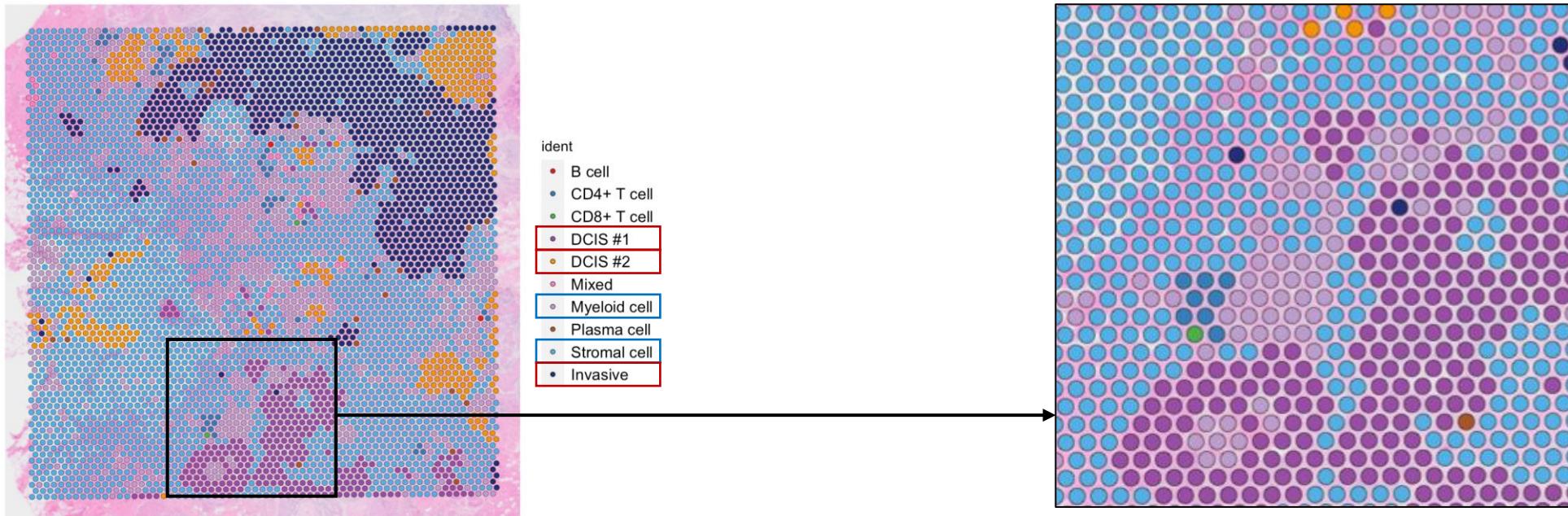
visium.breast$first_type = factor(visium.breast$first_type,
                                   levels = c("B cell", "CD4+ T cell", "CD8+ T cell",
                                             "DCIS #1", "DCIS #2", "Mixed", "Myeloid cell",
                                             "Plasma cell", "Stromal cell", "Invasive"))
```

Load data

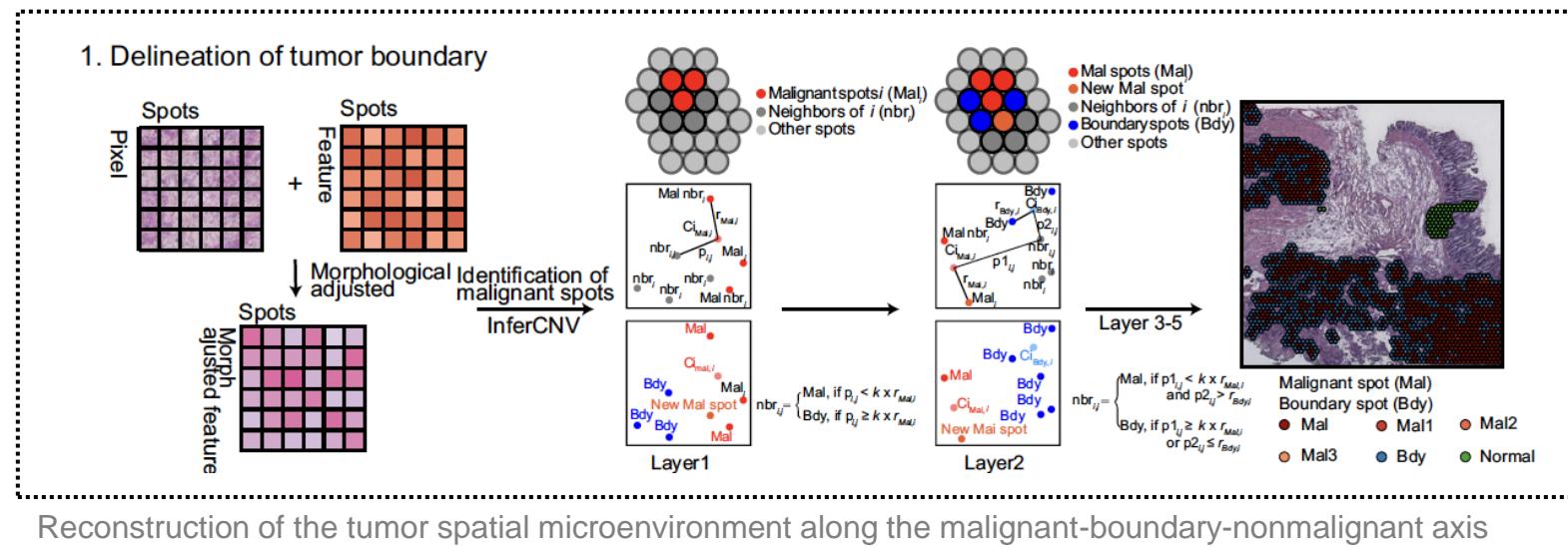
```
SpatialDimPlot(visium.breast, label = F, cols = colors)
```

There are many myeloid cells and stromal cells around the tumor.

We focus on the cell-cell interaction that occurs in the **tumor layer**.



Cottrazm to delineate tumor boundary



Cottrazm process

- 1 Contrtrazm adjusted gene expression with morphological information
- 2 Immune-related signatures were used to score spots and select reference cluster.
- 3 InferCNV was used to assess CNV level for remained spots
- 4 Annotate malignant cluster
- 5 Find neighbor spots of tumor core
- 6 Decide malignant spots (Mal), boundary spot (Bdy), and non-malignant spots (nMal)

Load data

```
# Load tumor annotated visium data  
visium.tumor = readRDS('./R_object/Bioinfo2023_Breast_visium_TumorST.rds')  
visium.tumor = subset(visium.tumor, nCount_Spatial > 100)  
table(visium.tumor@meta.data$tumor_annotation)
```

We only need the boundary cells of the **tumor and their layers**.

```
> table(visium.tumor@meta.data$tumor_annotation)
```

| Bdy | Mal | nMal |
|-----|------|------|
| 531 | 1143 | 3314 |

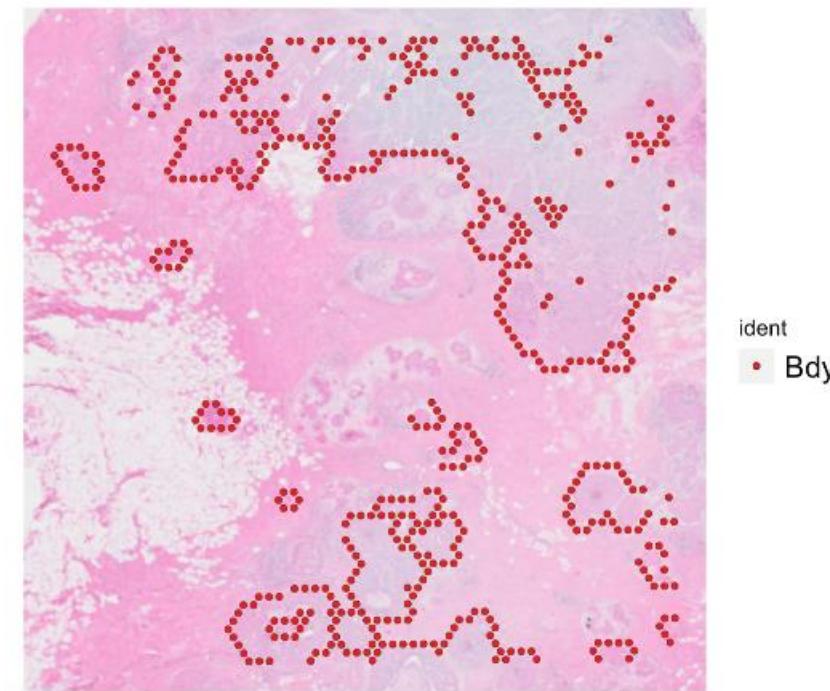
Load data

Subset only tumor boundaries

```
visium.breast@meta.data$tumor_annotation = visium.tumor@meta.data$tumor_annotation  
  
Idents(visium.breast) = visium.breast@meta.data$tumor_annotation  
visium.boundary = subset(visium.breast, idents = "Bdy")
```

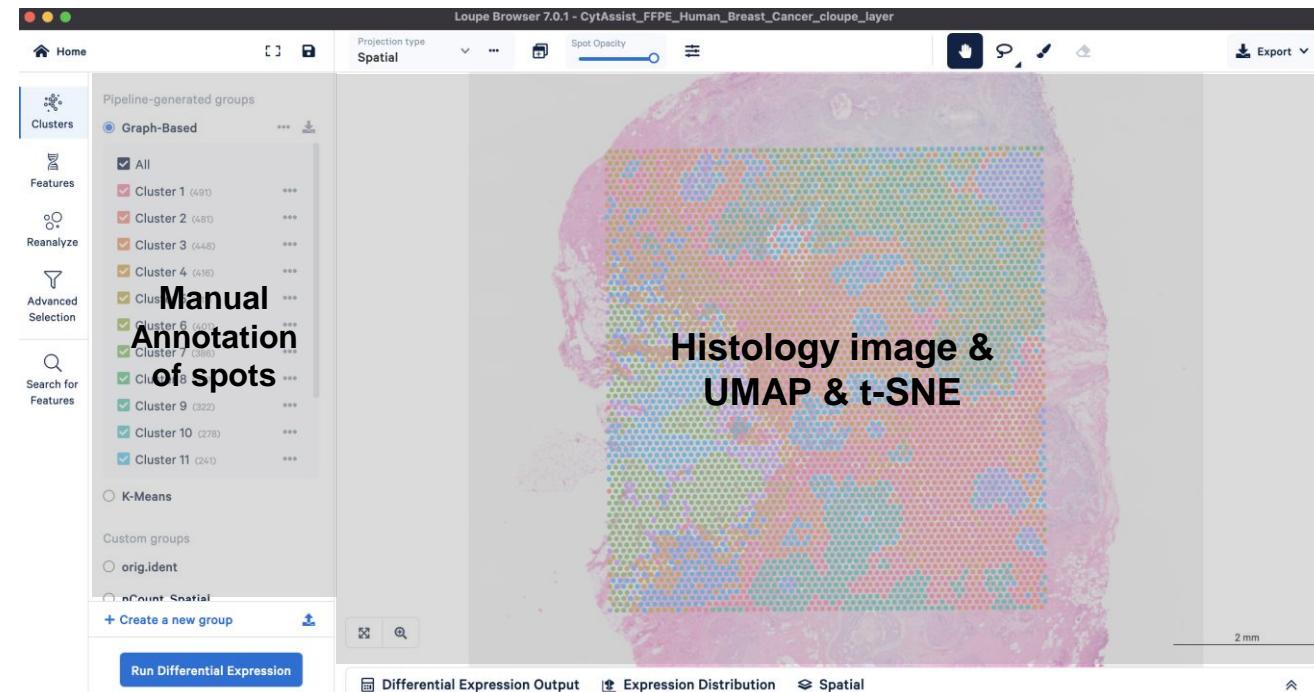
Load data

```
Idents(visium.boundary)= visium.boundary@meta.data$tumor_annotation  
names(colors) = "Bdy"  
  
SpatialDimPlot(visium.boundary, label = F, cols = colors)
```



What is a Loupe Browser?

- Loupe Browser is a desktop application from 10x Genomics that allows to visualize gene expression data without having to write code.
- Align gene expression spots to histological images, look for marker gene expression, annotate populations, and cluster.
- The .cloupe file is the one that need to import into the Loupe Browser.



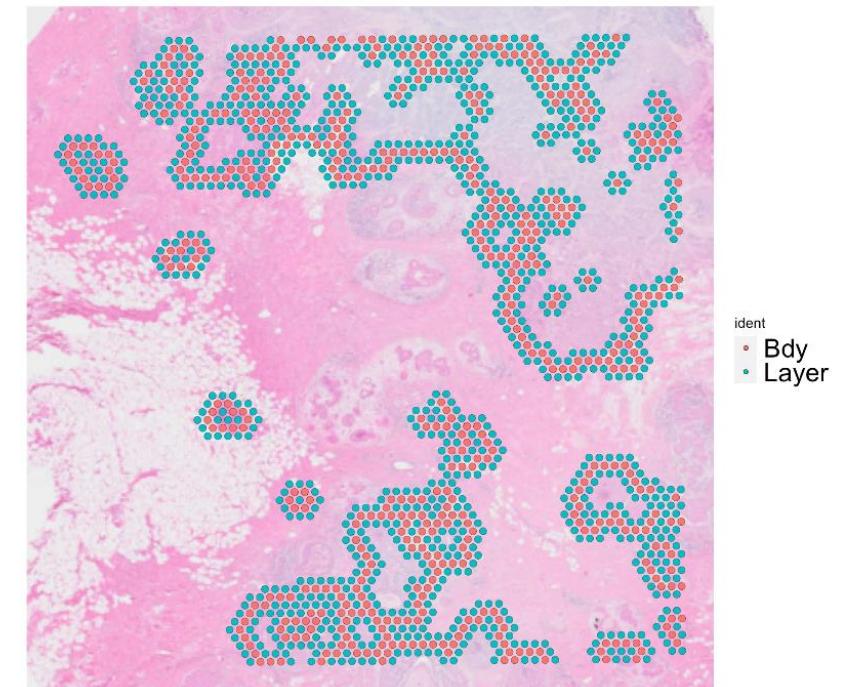
Load data

```
# Load layer annotated Loupe Browser data
loupe.data = read.csv("./Raw_file/LoupeBrowser/layer_annotation.csv",
                      header = T, row.names = 1)
table(loupe.data$layer_annotation)
```

Manually annotate the tumor layer using the Loupe browser.

```
> table(loupe.data$layer_annotation)
```

| Bdy | Layer | Mal | nMal |
|-----|-------|-----|------|
| 531 | 988 | 586 | 2883 |



Load data

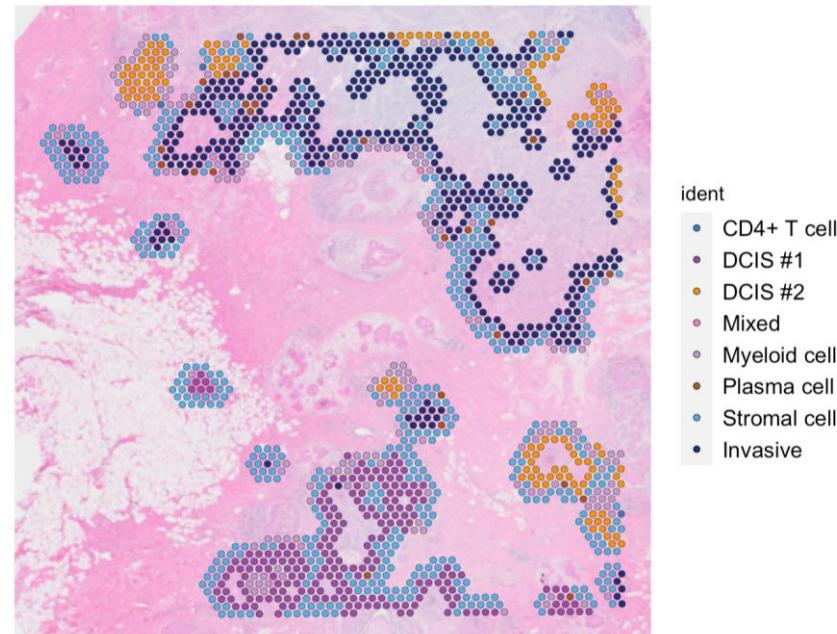
Subset only tumor boundaries and layers

```
visium.breast@meta.data = merge(visium.breast@meta.data, loupe.data,  
                                 by = "row.names")  
  
row.names(visium.breast@meta.data) = visium.breast@meta.data$Row.names  
  
Idents(visium.breast) = visium.breast@meta.data$layer_annotation  
  
tumor.bdy = subset(visium.breast, nCount_Spatial > 100,  
                   idents = c("Bdy", "Layer"))
```

Visualization of our data

```
Idents(tumor.bdy) = tumor.bdy$first_type  
names(colors) = c("B cell", "CD4+ T cell", "CD8+ T cell", "DCIS #1", "DCIS #2",  
                 "Mixed", "Myeloid cell", "Plasma cell", "Stromal cell", "Invasive")  
  
SpatialDimPlot(tumor.bdy, label = F, cols = colors)
```

There are 8 cell types from our final tumor boundary data.



Prepare input data for CellChat analysis

```
data.input = GetAssayData(tumor.bdy, slot = "data", assay = "SCT")
meta = data.frame(labels = Idents(tumor.bdy),
                   row.names = names(Idents(tumor.bdy)))  
  
# check the cell labels
unique(meta$labels)
```

```
> unique(meta$labels) # check the cell labels
[1] Stromal cell Invasive      DCIS #2      Myeloid cell DCIS #1      CD4+ T cell  Plasma cell  Mixed
Levels: CD4+ T cell DCIS #1 DCIS #2 Mixed Myeloid cell Plasma cell Stromal cell Invasive
```

Load spatial imaging information

```
# Load spatial imaging information to get the spot information
spatial.locs = GetTissueCoordinates(tumor.bdy, scale = NULL,
                                     cols = c("imagerow", "imagecol"))

scale.factors = jsonlite:::fromJSON(txt =
  "./Raw_file/visium/spatial/scalefactors_json.json")

scale.factors = list(spot.diameter = 65,
                     spot = scale.factors$spot_diameter_fullres,
                     fiducial = scale.factors$fiducial_diameter_fullres,
                     hires = scale.factors$tissue_hires_scalef,
                     lowres = scale.factors$tissue_lowres_scalef)
```

Create a CellChat object

Create a CellChat object for the downstream analysis

```
cellchat = createCellChat(object = data.input,  
                           meta = meta,  
                           group.by = "labels",  
                           datatype = "spatial",  
                           coordinates = spatial.locs,  
                           scale.factors = scale.factors)  
  
cellchat
```

> cellchat

An object of class CellChat created from a single dataset

18045 genes.

1519 cells.

CellChat analysis of spatial data! The input spatial locations are

| | x_cent | y_cent |
|--------------------|--------|--------|
| AACAGGATTCTAGTT-1 | 12365 | 11404 |
| AACAGGTTCACCGAAG-1 | 12682 | 11589 |
| AACAGTCCACGCGGTG-1 | 6464 | 14372 |
| AACATCTTAAGGCTCA-1 | 7098 | 14560 |
| AACCAATCTGGTTGGC-1 | 12824 | 13691 |
| AACCACAACTGATT-1 | 13934 | 13973 |

Set the ligand-receptor interaction database

```
# Load CellChat DB
CellChatDB = CellChatDB.human
cellchat.gene = as.data.frame(CellChatDB.human$geneInfo$Symbol)
colnames(cellchat.gene) = "gene"

# Load Xenium gene panel
xenium.gene = read.csv("./Raw_file/xenium/Xenium_FFPE_Human_Breast_Cancer_Rep1_gene_groups.csv")
```

CellChatDB : Manually curated database of literature-supported ligand-receptor interactions in both **human and mouse**.

Since our toy data is a human breast 10x visium data, we load **CellChatDB.human**.

Xenium In Situ Datasets : **313** genes chosen to explore a Xenium In Situ dataset from human breast cancer FFPE section.

Set the ligand-receptor interaction database

```
# Filter CellChat DB by Xenium gene
overlap.gene = merge(cellchat.gene, xenium.gene, by = "gene")

CellChatDB$interaction =
  CellChatDB$interaction[CellChatDB$interaction$ligand %in% overlap.gene$gene &
    CellChatDB$interaction$receptor %in% overlap.gene$gene,]

cellchat@DB = CellChatDB
```

Preprocess of the expression data for cell-cell communication analysis

```
# Subset the expression data of signaling genes for saving computation cost  
cellchat = subsetData(cellchat)
```

```
# Identify over-expressed ligands or receptors in one cell group  
cellchat = identifyOverExpressedGenes(cellchat)
```

```
# Identify over-expressed ligand-receptor interactions if either ligand or receptor is over-expressed  
cellchat = identifyOverExpressedInteractions(cellchat)
```

Compute the communication probability and infer cellular communication network

Infers the biologically significant cell-cell communication with permutation test

```
cellchat = computeCommunProb(cellchat, type = "triMean", distance.use = TRUE,  
                             interaction.length = 200, scale.distance = 0.1)
```

Compute the communication probability and infer cellular communication network

Check the number of spots of a cell type

```
cellchat@meta$labels %>% table() %>% sort()
```

Filter cell-cell communication if there are only few number of spots in certain cell types

```
cellchat = filterCommunication(cellchat, min.cells = 10)
```

```
> cellchat@meta$labels %>% table() %>% sort()
```

.

| | | | | | | |
|-------|-------------|-------------|---------|--------------|---------|-----------------------|
| Mixed | CD4+ T cell | Plasma cell | DCIS #2 | Myeloid cell | DCIS #1 | Invasive Stromal cell |
| 1 | 6 | 33 | 137 | 177 | 209 | 465 |

Infer the cell-cell communication at a signaling pathway level

```
# Computes the communication probability on signaling pathway level  
cellchat = computeCommunProbPathway(cellchat)
```

CellChat can systematically classify ligand-receptor pairs into functionally related signaling pathways.

The inferred intercellular communication network of each ligand-receptor pair and each signaling pathway is stored in the slot ‘net’ and ‘netP’, respectively.

Calculate the aggregated cell-cell communication network

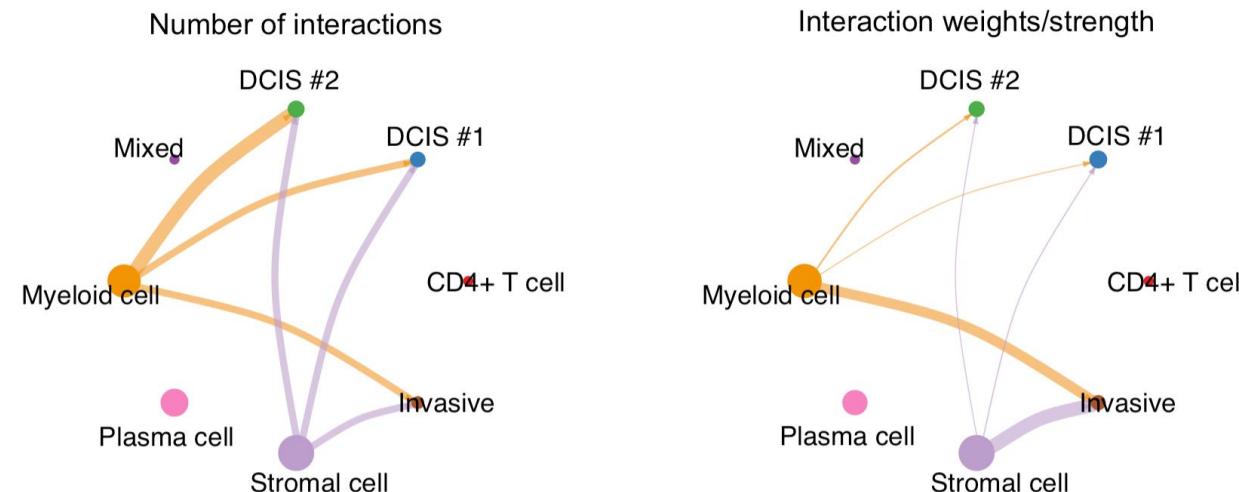
```
# Calculate the aggregated cell-cell communication network
cellchat = aggregateNet(cellchat)

#saveRDS(cellchat, file = './R_object/Bioinfo2023_Breast_CellChat.rds')

# read RDS file if computeCommunProb() takes too much time (optional)
cellchat = readRDS('./R_object/Bioinfo2023_Breast_CellChat.rds')
```

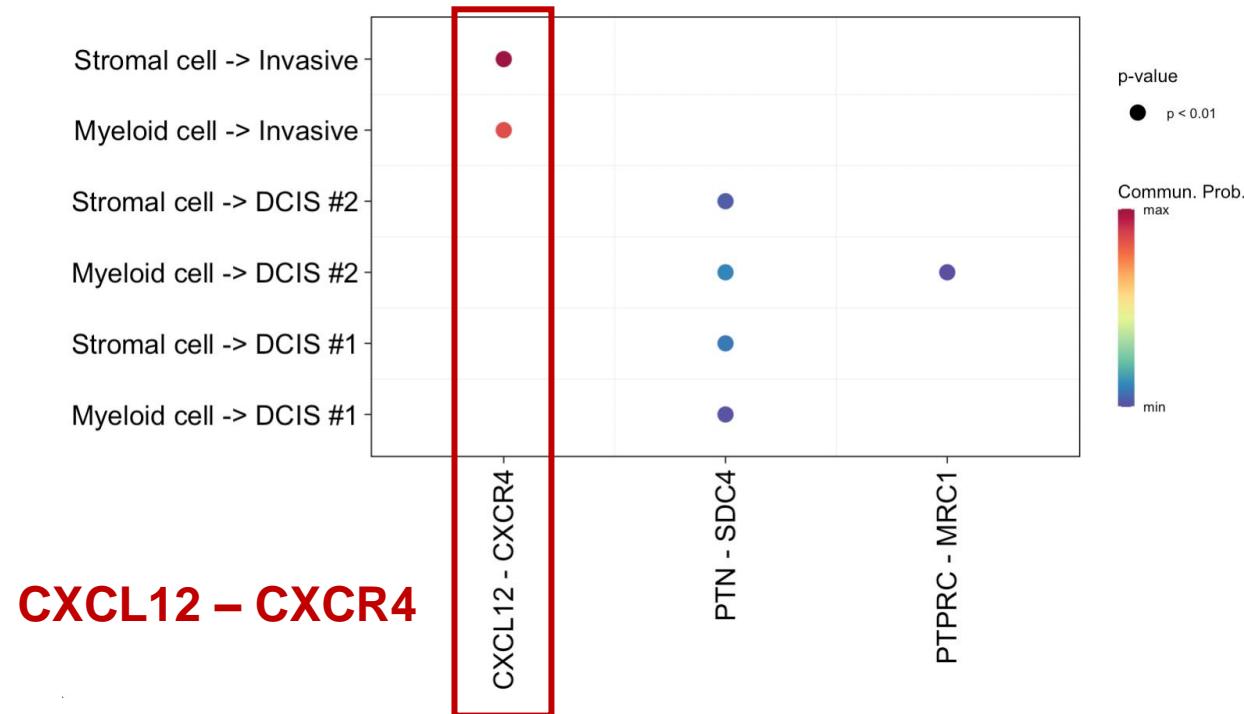
Visualization of the aggregated cell-cell communication network

```
netVisual_circle(cellchat@net$count, vertex.weight = rowSums(cellchat@net$count),  
                 sources.use = c("Stromal cell", "Myeloid cell"),  
                 targets.use = c("DCIS #1", "DCIS #2", "Invasive"),  
                 title.name = "Number of interactions")  
  
netVisual_circle(cellchat@net$weight, vertex.weight = rowSums(cellchat@net$weight),  
                 sources.use = c("Stromal cell", "Myeloid cell"),  
                 targets.use = c("DCIS #1", "DCIS #2", "Invasive"),  
                 title.name = "Interaction weights/strength")
```



Identify ligand-receptor pairs between cell types

```
CellChat:::netVisual_bubble(cellchat, sources.use = c("Stromal cell", "Myeloid cell"),
                            targets.use = c("DCIS #1", "DCIS #2", "Invasive"),
                            remove.isolate = FALSE, angle.x = 90, thresh = 0.05) +
coord_flip()
```



Compute the network centrality scores

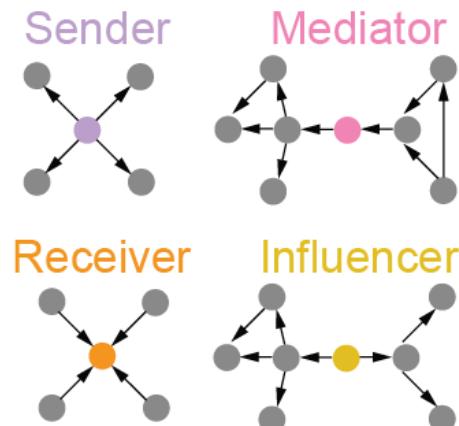
Compute the network centrality scores

```
cellchat = netAnalysis_computeCentrality(cellchat, net.name = "CXCL12-CXCR4")
```

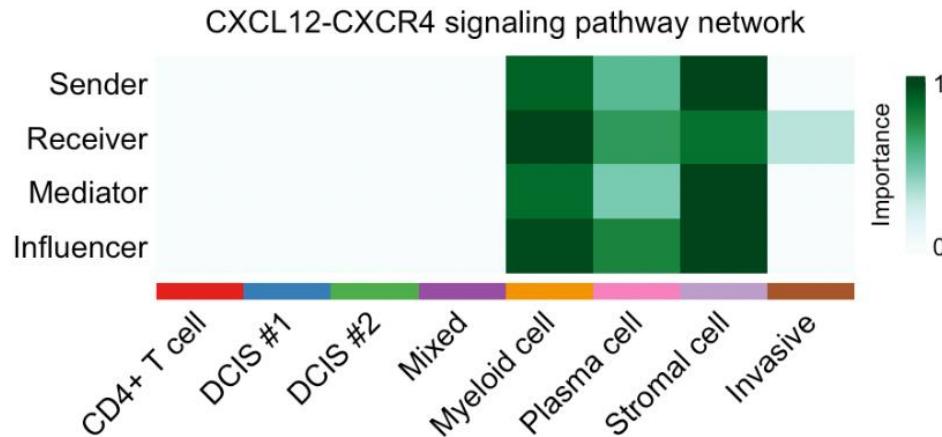
Visualize the centrality score

```
netAnalysis_signalingRole_network(cellchat, signaling = "CXCL12-CXCR4",
                                    width = 8, height = 2.5, font.size = 10)
```

Visualize the computed centrality scores using heatmap, allowing ready identification of major signaling roles of cell groups.



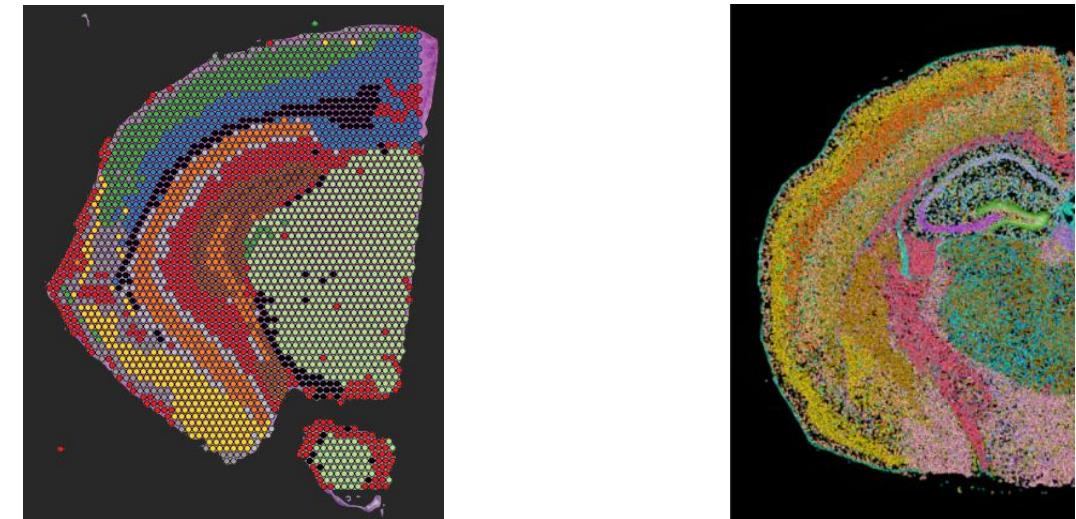
<https://github.com/jinworks/CellChat>



4

Xenium In Situ

Differences between Visium and Xenium



| | 10x Visium | 10x Xenium |
|-----------------------------|--------------------------|--|
| Number of genes | Tens of thousands | Hundreds to thousands |
| Spatial resolution | Lower resolution (~55µm) | Higher subcellular resolution (~200nm) |
| Cell-cell boundaries | Not defined | Defined with cellular segmentation |

Load and preprocess the dataset

Load the Xenium data

```
s1r1 = LoadXenium('Raw_file/xenium/Xenium_FFPE_Human_Breast_Cancer_Rep1_outs', fov = 'fov')
```

Remove cells with 0 counts

```
S1r1 = subset(s1r1, subset = nCount_Xenium > 0)
```

Add metadata

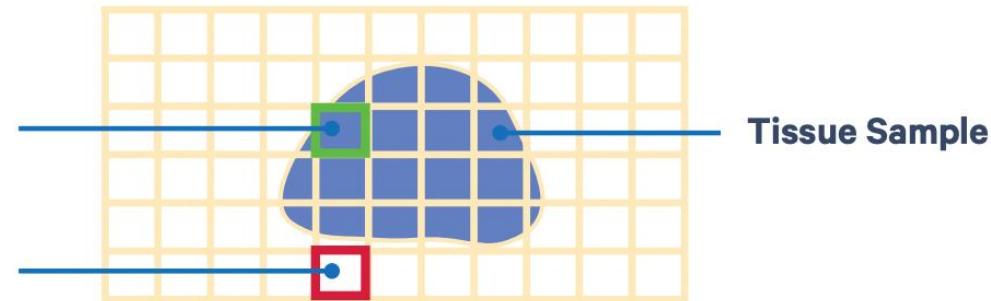
```
s1r1@meta.data$cells = 'cells'
```

Field of View

One box is considered one field of view



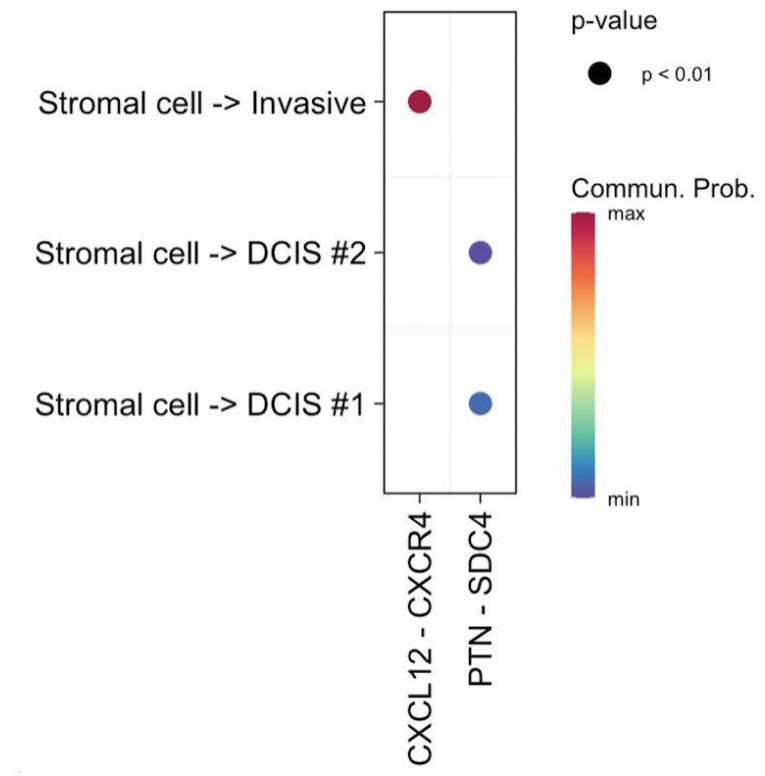
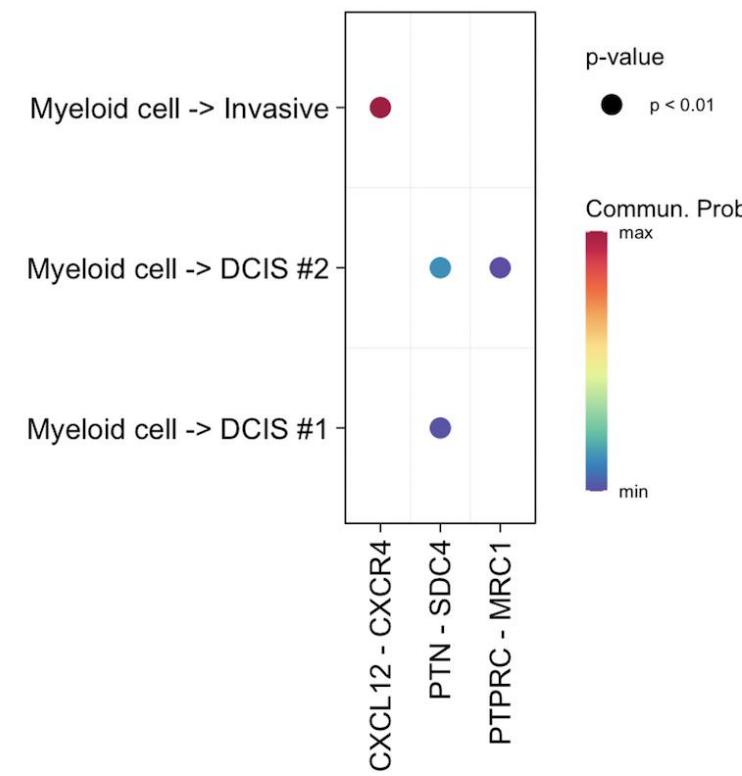
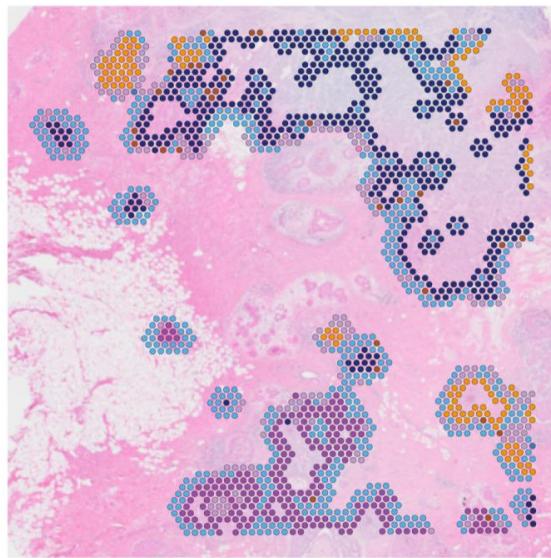
DO NOT select empty FOVs



https://cdn.10xgenomics.com/image/upload/v1694469210/support-documents/CG000584_Xenium_Analyzer_UserGuide_RevC.pdf

Intercellular communication inferred by cellchat

- Myeloid / Stromal cells - tumor cells (CXCL12 - CXCR4, PTN - SDC4)



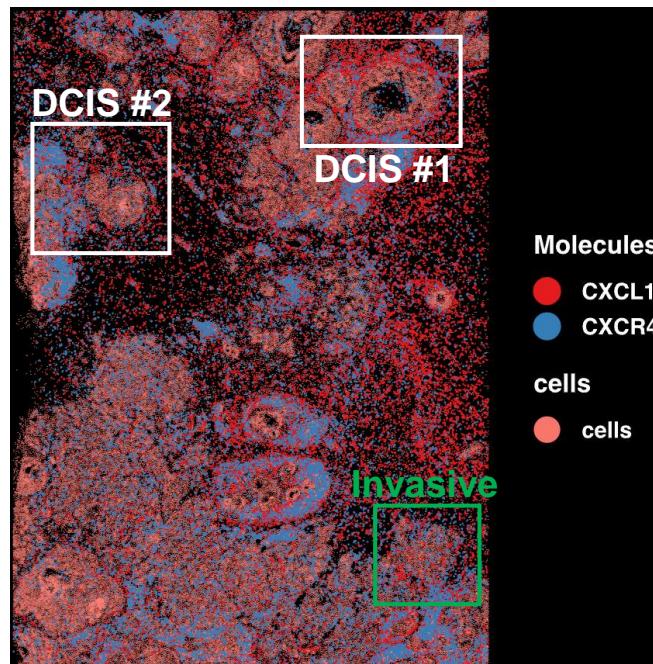
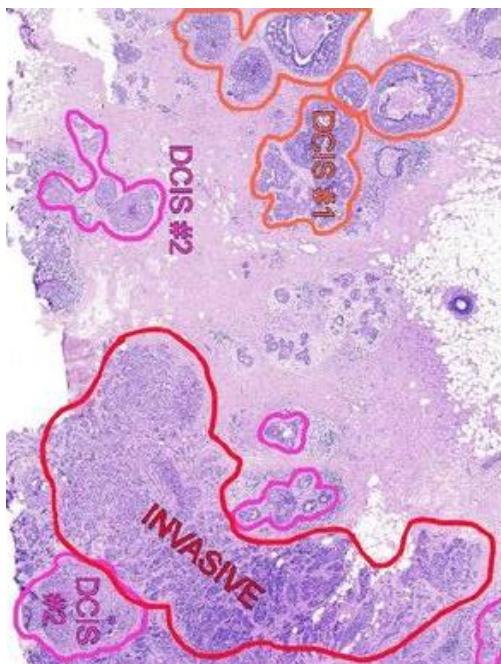
Visualize the expression level of CXCL12 and CXCR4

Plot the positions of CXCL12 and CXCR4

```
ImageDimPlot(s1rl, fov = "fov", molecules = c("CXCL12", "CXCR4"), group.by = 'cells', nmols = 20000)
```

Visualize the expression level of CXCL12 and CXCR4

```
ImageFeaturePlot(s1rl, features = c("CXCL12", "CXCR4"), max.cutoff = c(15, 3), size = 0.5, cols = c("white", "red"))
```



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

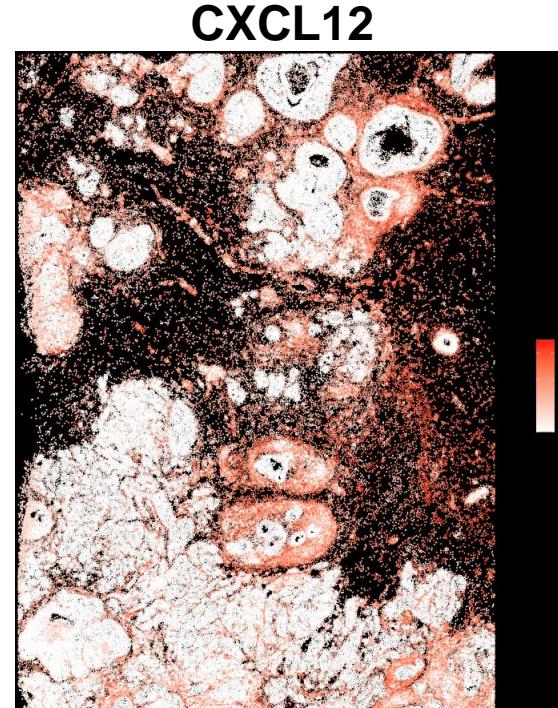
16

17

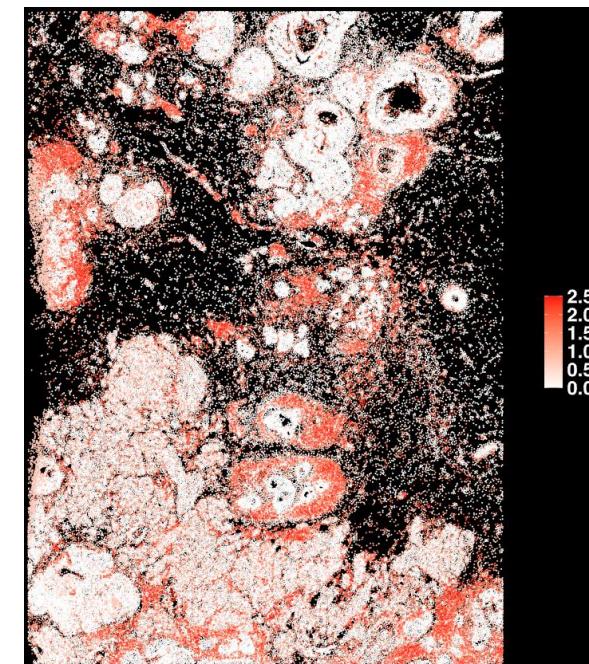
18

19

20



CXCL12



CXCR4

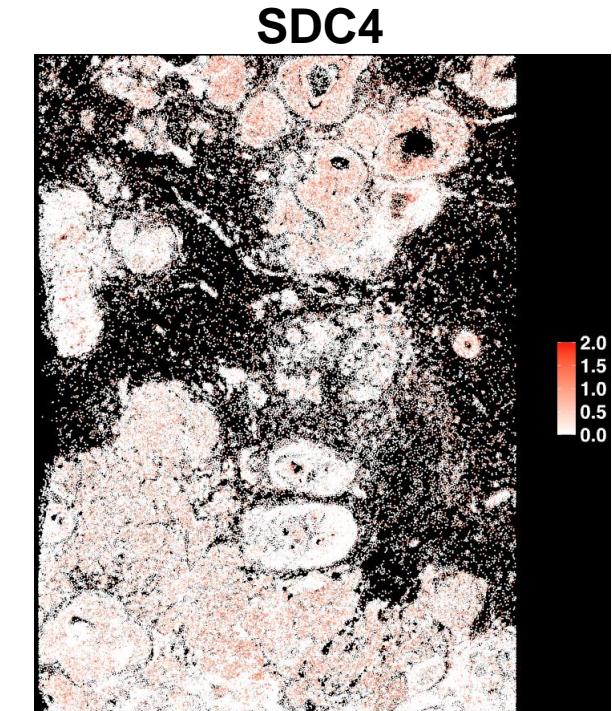
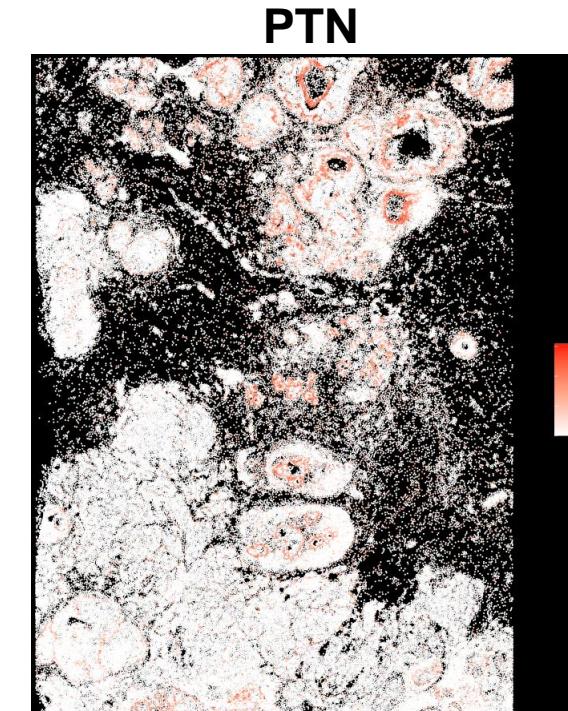
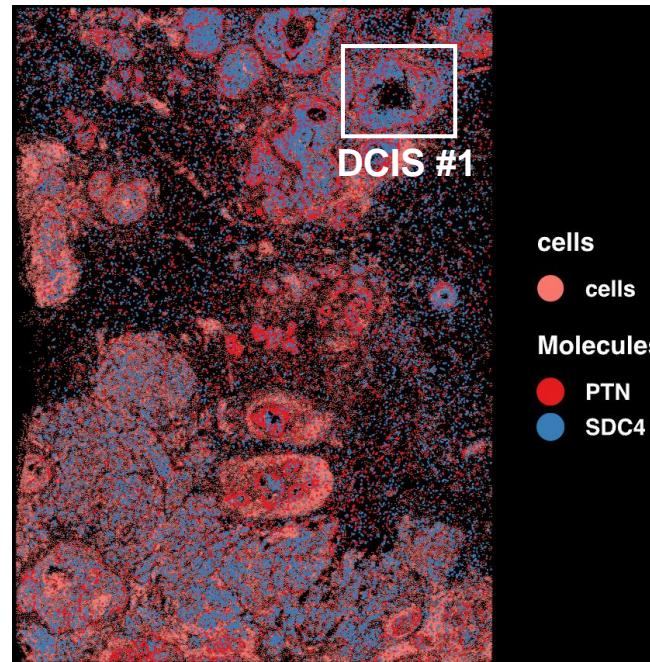
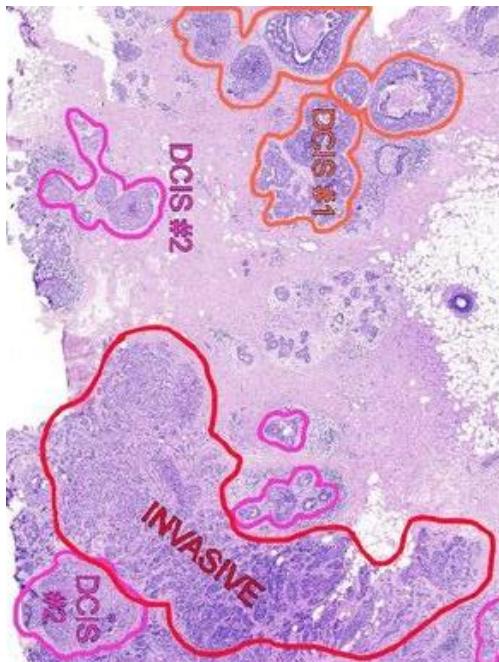
Visualize the expression level of PTN and SDC4

Plot the positions of PTN and SDC4

```
ImageDimPlot(s1rl, fov = "fov", molecules = c("PTN", "SDC4"), group.by = 'cells', nmols = 20000)
```

Visualize the expression level of PTN and SDC4

```
ImageFeaturePlot(s1rl, features = c("PTN", "SDC4"), max.cutoff = c(8, 8), size = 0.5, cols = c("white", "red"))
```



Zoom in on the PTN – SDC4 binding area

Increase your RAM usage (8GB)

```
options(future.globals.maxSize = 8000 * 1024^2)
```

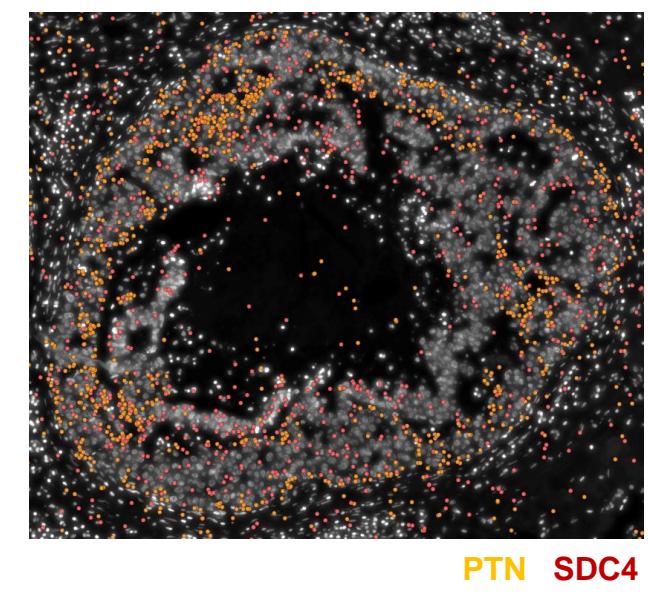
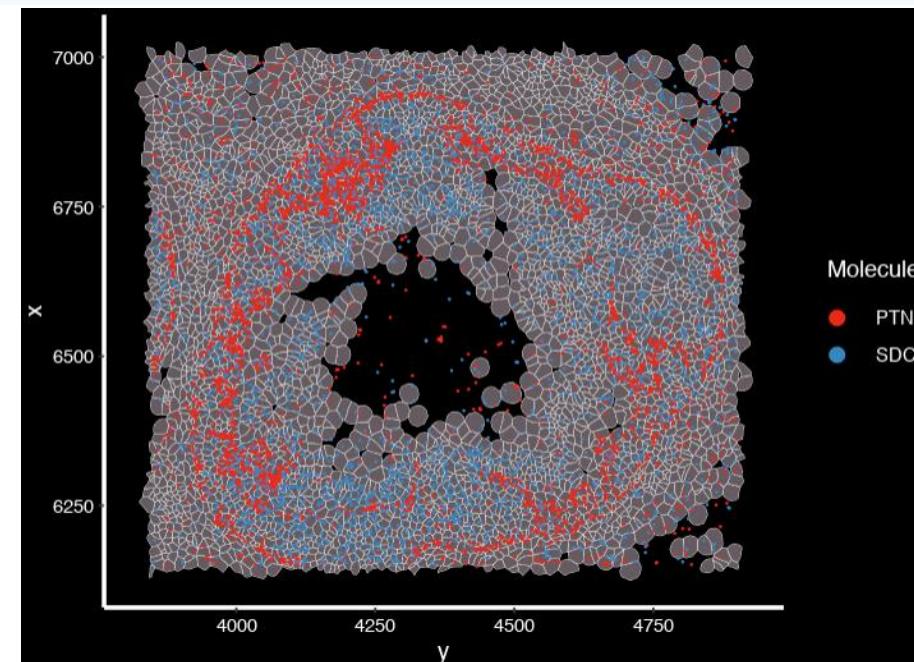
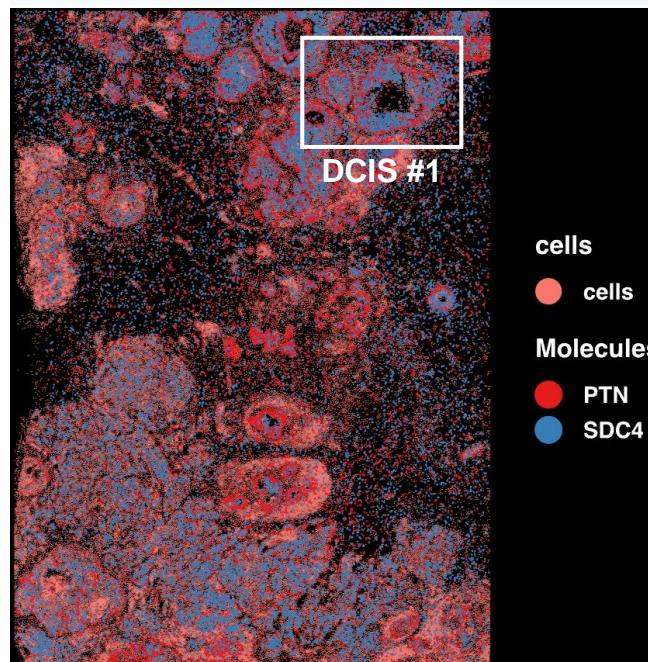
Define cropped area

```
cropped.coords = Crop(s1rl1[["fov"]], x = c(3850, 4900), y = c(6150, 7000), coords = "plot")  
s1rl1[["zoom"]] = cropped.coords
```

Visualize cropped area with cell segmentations & selected molecules

```
DefaultBoundary(s1rl1[["zoom"]]) = "segmentation"
```

```
ImageDimPlot(s1rl1, fov = "zoom", axes = TRUE, border.color = "white", border.size = 0.1, cols =  
"polychrome", coord.fixed = FALSE, molecules = c("PTN", "SDC4"), nmols = 10000, group.by = 'cells')
```

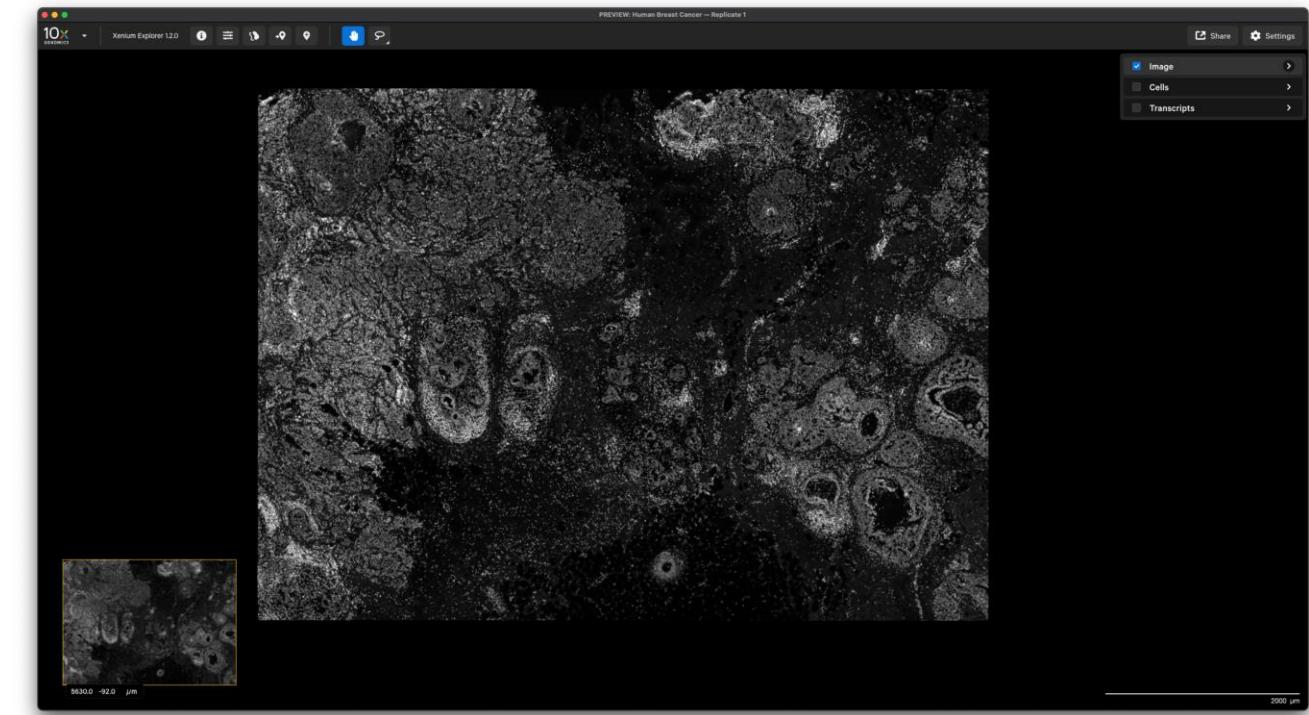


Visualize RNA transcript localization in tissue using Xenium Explorer

10x genomics Xenium Explorer

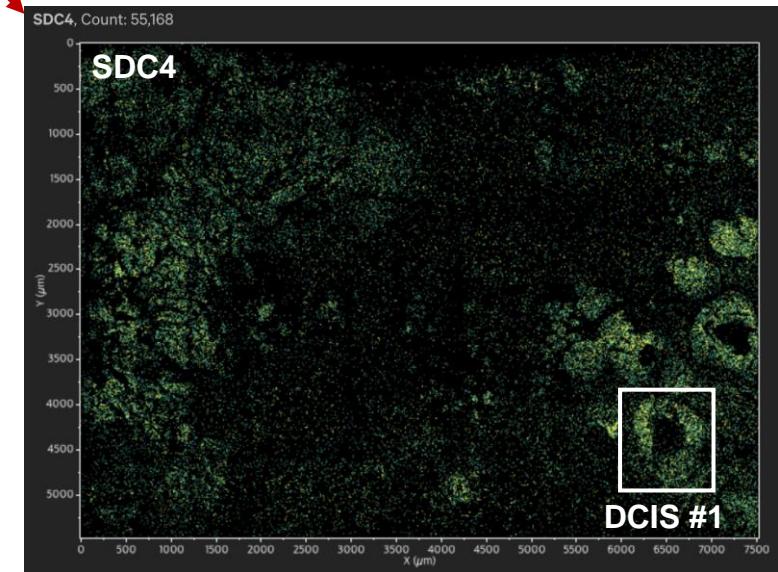
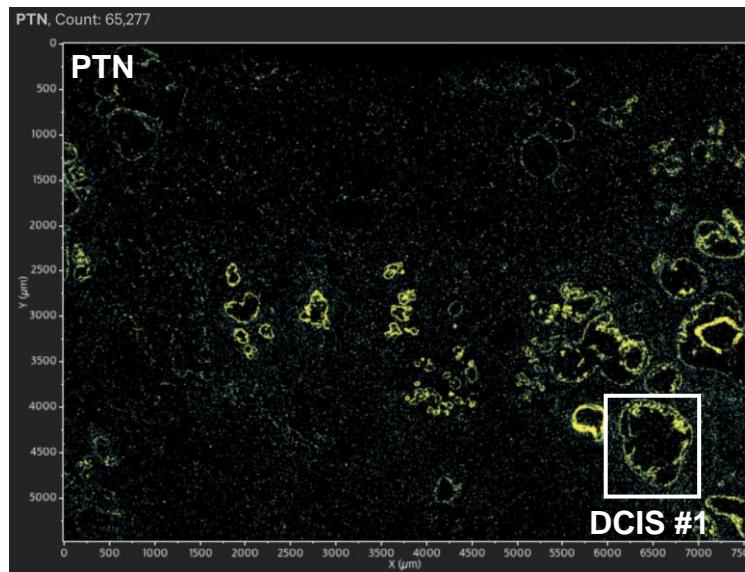
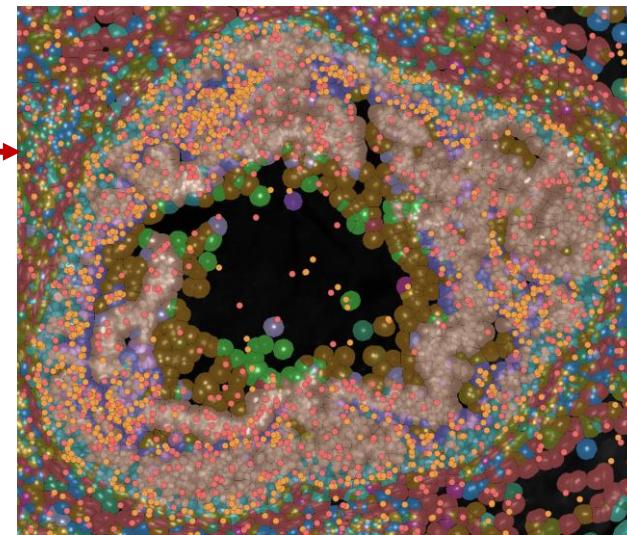
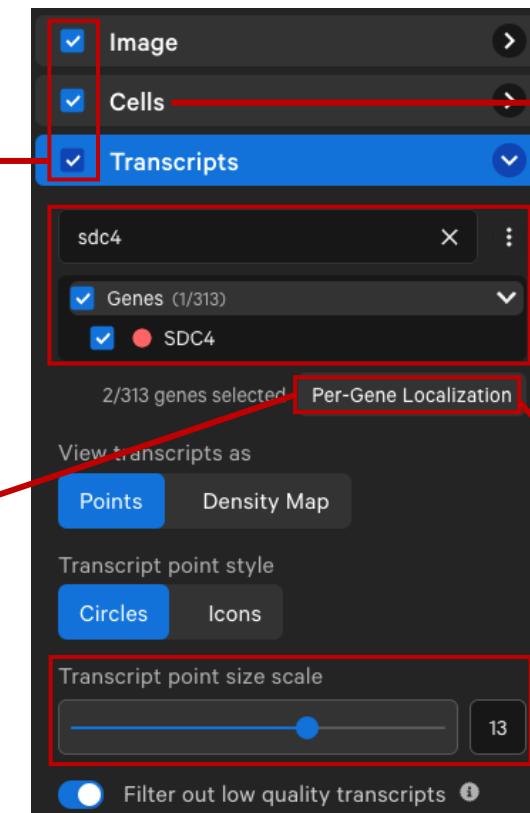
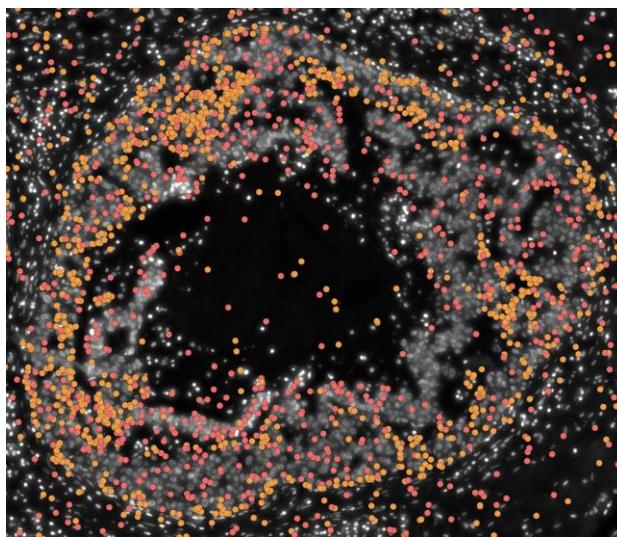
The screenshot shows the 'Download Center' section of the Xenium Explorer website. It features two main download links: 'Download for Windows' (file size 112 MB) and 'Download for Mac OS' (file size 145 MB). Below these links, there is a section titled 'Xenium Explorer 1.3.0 highlights' which lists several key features of the software.

The screenshot shows the Xenium Explorer application window. At the top right, there is a blue button labeled 'Open New File'. A red arrow points from the text 'Raw_file/xenium/Xenium_FFPE_Human_Breast_Cancer_Rep1_outs/experiment.xenium' to this button. The main area displays a preview image of a tissue sample with the text 'PREVIEW: Human Breast Cancer' and 'Region Replicate 1 | Cells 167,780 | Panel Breast Cancer, Tumor Microenvironment'.



Visualize RNA transcript localization in tissue using Xenium Explorer

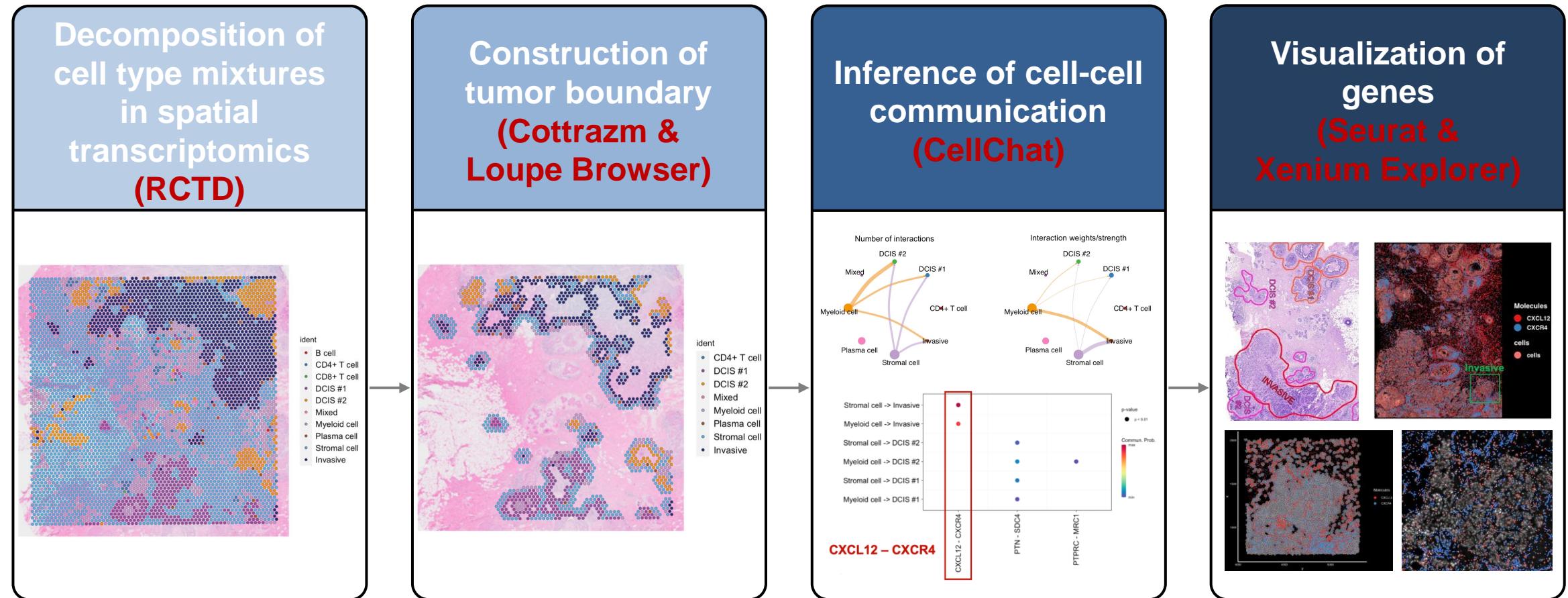
PTN SDC4



5

Summary

Summary



References

- **Seurat**

Hao Y, Stuart T, Kowalski MH, Choudhary S, Hoffman P, Hartman A, Srivastava A, Molla G, Madad S, Fernandez-Granda C, Satija R (2023). “Dictionary learning for integrative, multimodal and scalable single-cell analysis.” *Nature Biotechnology*. doi:10.1038/s41587-023-01767-y

- **RCTD**

Cable, D.M., Murray, E., Zou, L.S. *et al.* Robust decomposition of cell type mixtures in spatial transcriptomics. *Nat Biotechnol* **40**, 517–526 (2022).

- **CellChat**

Suoqin Jin *et al.*, Inference and analysis of cell-cell communication using CellChat, *Nature Communications* 2021

- **Cottrazm**

Xun, Z., Ding, X., Zhang, Y. *et al.* Reconstruction of the tumor spatial microenvironment along the malignant-boundary-nonmalignant axis. *Nat Commun* **14**, 933 (2023).

- **Loupe Browser**

<https://www.10xgenomics.com/support/software/loupe-browser>

- **Xenium**

https://satijalab.org/seurat/articles/spatial_vignette_2#mouse-brain-10x-genomics-xenium-in-situ
<https://www.10xgenomics.com/support/software/xenium-explorer>

Q & A

jungminchoi@korea.ac.kr
kwangminyoo@korea.ac.kr