

KSBi-BIML 2023

Bioinformatics & Machine Learning(BIML)
Workshop for Life Scientists, Data Scientists,
and Bioinformaticians

생물정보학 & 머신러닝 워크샵 (온라인)

Single cell analysis
(spatial transcriptomics)

최정민 _ 고려대학교



본 강의 자료는 한국생명정보학회가 주관하는 BIML 2023 워크샵 온라인 수업을 목적으로 제작된 것으로 해당 목적 이외의 다른 용도로 사용할 수 없음을 분명하게 알립니다.

이를 다른 사람과 공유하거나 복제, 배포, 전송할 수 없으며 만약 이러한 사항을 위반할 경우 발생하는 **모든 법적 책임은 전적으로 불법 행위자 본인에게 있음을 경고**합니다.

KSBi-BIML 2023

Bioinformatics & Machine Learning (BIML) Workshop for Life Scientists, Data Scientists, and Bioinformaticians

안녕하십니까?

한국생명정보학회가 개최하는 동계 교육 워크숍인 BIML-2023에 여러분을 초대합니다. 생명정보학 분야의 연구자들에게 최신 동향의 데이터 분석기술을 이론과 실습을 겸비해 전달하고자 도입한 전문 교육 프로그램인 BIML 워크숍은 2015년에 시작하여 올해로 9차를 맞이하게 되었습니다. 지난 2년간은 심각한 코로나 대유행으로 인해 아쉽게도 모든 강의가 온라인으로 진행되어 현장 강의에서만 가능한 강의자와 수강생 사이에 다양한 소통의 기회가 없음에 대한 아쉬움이 있었습니다. 다행히도 최근 사회적 거리두기 완화로 현장 강의를 가능해져 올해는 현장 강의를 재개함으로써 온라인과 현장 강의의 장점을 모두 갖춘 프로그램을 구성할 수 있게 되었습니다.

BIML 워크숍은 전통적으로 크게 인공지능과 생명정보분석 두 개의 분야로 구성되었습니다. 올해 AI 분야에서는 최근 생명정보 분석에서도 응용이 확대되고 있는 다양한 심층학습(Deep learning) 기법들에 대한 현장 강의를 진행될 예정이며, 관련하여 심층학습을 이용한 단백질구조예측, 유전체 분석, 신약개발에 대한 이론과 실습 강의를 함께 제공할 예정입니다. 또한 싱글셀오믹스 분석과 메타유전체분석 현장 강의는 많은 연구자의 연구 수월성 확보에 큰 도움을 줄 것으로 기대하고 있습니다. 이외에 다양한 생명정보학 분야에 대하여 30개 이상의 온라인 강좌가 개설되어 제공되며 온라인 강의의 한계를 극복하기 위해서 실시간 Q&A 세션 또한 마련했습니다. 특히 BIML은 각 분야 국내 최고 전문가들의 강의로 구성되어 해당 분야의 기초부터 최신 연구 동향까지 포함하는 수준 높은 내용의 강의를 될 것입니다.

이번 BIML-2023을 준비하기까지 너무나 많은 수고를 해주신 BIML-2023 운영위원회의 남진우, 우현구, 백대현, 정성원, 정인경, 장혜식, 박종은 교수님과 KOBIC 이병욱 박사님께 커다란 감사를 드립니다. 마지막으로 부족한 시간에도 불구하고 강의 부탁을 흔쾌히 허락하시고 훌륭한 현장 강의와 온라인 강의를 준비하시는데 노고를 아끼지 않으신 모든 연사분께 깊은 감사를 드립니다.

2023년 2월

한국생명정보학회장 이 인 석

Single cell analysis (spatial transcriptomics)

최근 유전체 분석 기술의 지속적인 발전으로 단일세포 수준에서의 세포간의 이질성을 확인할 수 있게 됨에 따라 다양한 생물학적 기전에 대한 분자적인 수준에서의 이해가 높아지고 있다.

이와 더불어, 공간전사체 분석 기술의 등장으로 세포들의 공간적 분포나 맥락을 분석에 고려할 수 있게 되어 보다 복잡한 생물학적인 기전에 대한 이해를 도전할 수 있게 되었다.

본 워크샵에서는 일반에 공개된 사람의 배외측 전전두피질 조직 10x Visium 데이터를 활용해 전반적이고 심층적인 공간전사체 분석을 진행, 그에 대한 생물학적 해석을 하는 것을 목표로 한다.

강의는 다음의 내용을 포함한다:

- Introduction to Spatially Resolved Transcriptomics (SRT)
- SRT preprocessing
- SRT analysis
- SRT workflow

* 교육생준비물:

노트북 (메모리 8GB 이상, 디스크 여유공간 30GB 이상)

* 강의 난이도: 초급

* 강의: 최정민 교수 (고려대학교 의과학과 컴퓨터공학부) / 천하림, 김지현, 박주영 조교

Curriculum Vitae

Speaker Name: Jungmin Choi Ph.D.



► Personal Info

Name Jungmin Choi
Title Associate Professor
Affiliation Korea University

► Contact Information

Address 73, Goryeodae-ro, Seongbuk-gu, Seoul 02841, South Korea
Email jungminchoi@korea.ac.kr
Phone Number 010-2120-9874

Research Interest

Genetics, genomics, computational biology

Educational Experience

2004 B.S. in Chemistry, Yonsei university, Korea
2012 Ph.D. in Genetics, University of Maryland, USA

Professional Experience

2013-2018 Postdoctoral research fellow, Yale University, USA
2018-2019 Research Associate, Rockefeller University, USA

Selected Publications (5 maximum)

1. Lim VY*, Feng X*, Miao R, Zehentmeier S, Ewing-Crystal N, Lee M, Tumanov AV, Oh JE, Iwasaki A, Wang A, Choi J§, Pereira JP§. Mature B cells and Mesenchymal Stem Cells control emergency myelopoiesis. *J Exp Med*. 2022 in press.
2. Manavella DD, McNamara B, Harold J, Bellone S, Hartwich TMP, Yang-Hartwich Y, Mutlu L, Zipponi M, Demirkiran C, Verzosa MS, Altwerger G, Ratner E, Huang GS, Clark M, Andikyan V, Azodi M, Schwartz PE, Dottino PR, Choi J, Alexandrov LB, Buza N, Hui P, Santin AD. Ovarian and uterine carcinosarcomas are sensitive in vitro and in vivo to Elimusertib, a novel ataxia-telangiectasia and Rad3-related (ATR) kinase inhibitor. *Gynecol Oncol*. 2022 in press.
3. Harold J, Bellone S, Manavella DD, Mutlu L, McNamara B, Hartwich TMP, Zipponi M, Yang-Hartwich Y, Demirkiran C, Verzosa MS, Choi J, Dong W, Buza N, Hui P, Altwerger G, Huang GS, Andikyan V, Clark M, Ratner E, Azodi M, Schwartz PE, Santin AD. Elimusertib (BAY1895344), a novel ATR inhibitor, demonstrates in vivo activity in ATRX mutated models of uterine leiomyosarcoma. *Gynecol Oncol*. 2022 Nov 25;168:157-165. doi: 10.1016/j.ygyno.2022.11.014. Epub ahead of print. PMID: 36442427.
4. Kim Y, Kim C, Lee H, Kim M, Zheng H, Lim JY, Yun HI, Jeon M, Choi J, Hwang SW. Gpr83 Tunes Nociceptor Function, Controlling Pain. *Neurotherapeutics*. 2022 Nov 9. doi: 10.1007/s13311-022-01327-3. Epub ahead of print. PMID: 36352334.
5. Gauhar Z, Tejwani L, Abdullah U, Saeed S, Shafique S, Badshah M, Choi J, Dong W, Nelson-Williams C, Lifton RP, Lim J, Raja GK. A Novel Missense Mutation in ERCC8 Co-Segregates with Cerebellar Ataxia in a Consanguineous Pakistani Family. *Cells*. 2022 Sep 30;11(19):3090. doi: 10.3390/cells11193090. PMID: 36231052; PMCID: PMC9564319.

Spatially Resolved Transcriptomics (SRT)

Brief Overview

Jungmin Choi, Ph.D.

Department of Biomedical Sciences
Korea University, College of Medicine

Contents covered today

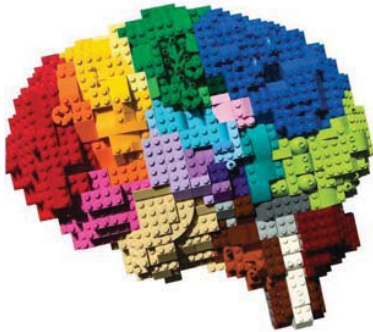
- **Introduction**

- A broad overview of single-cell data and experimental spatially resolved techniques

- **Computational methodology and frameworks**

- Different flavors of currently available spatially resolved data analysis methods

Single cells help gain insight into complexity of biology



Brain tissue



Bulk genomics

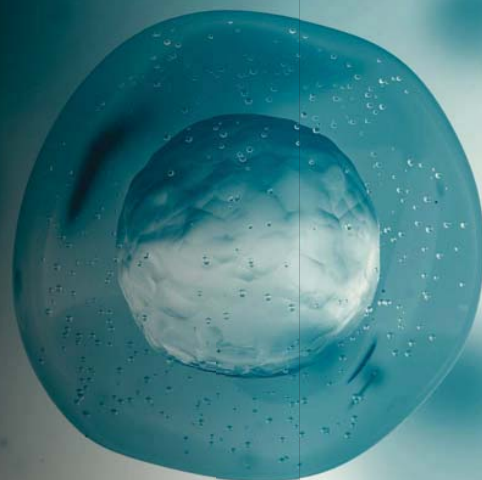


Single-cell genomics

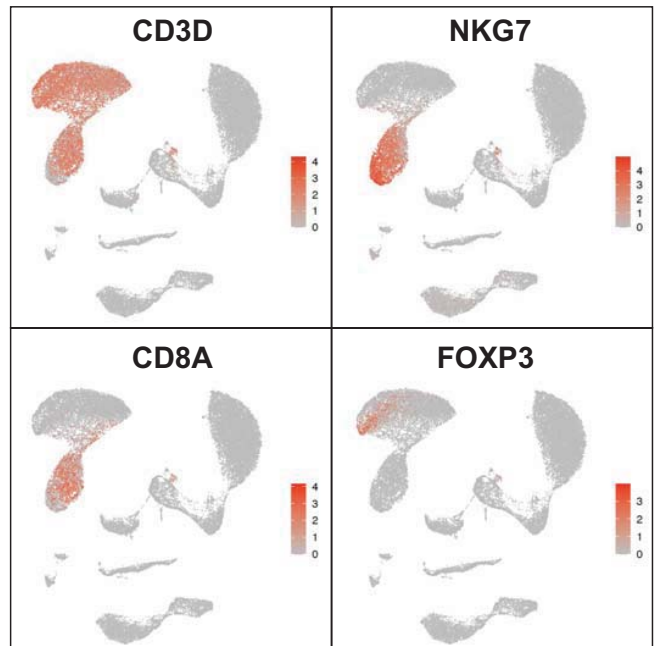
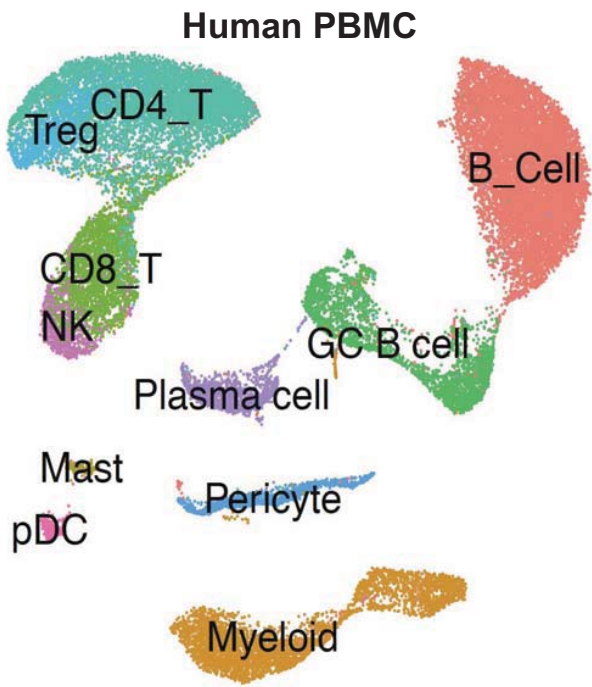
The average behavior measured in millions of cells (bulk genomics) does not necessarily reflect the behavior in individual cells

Image credit to Bo Xia

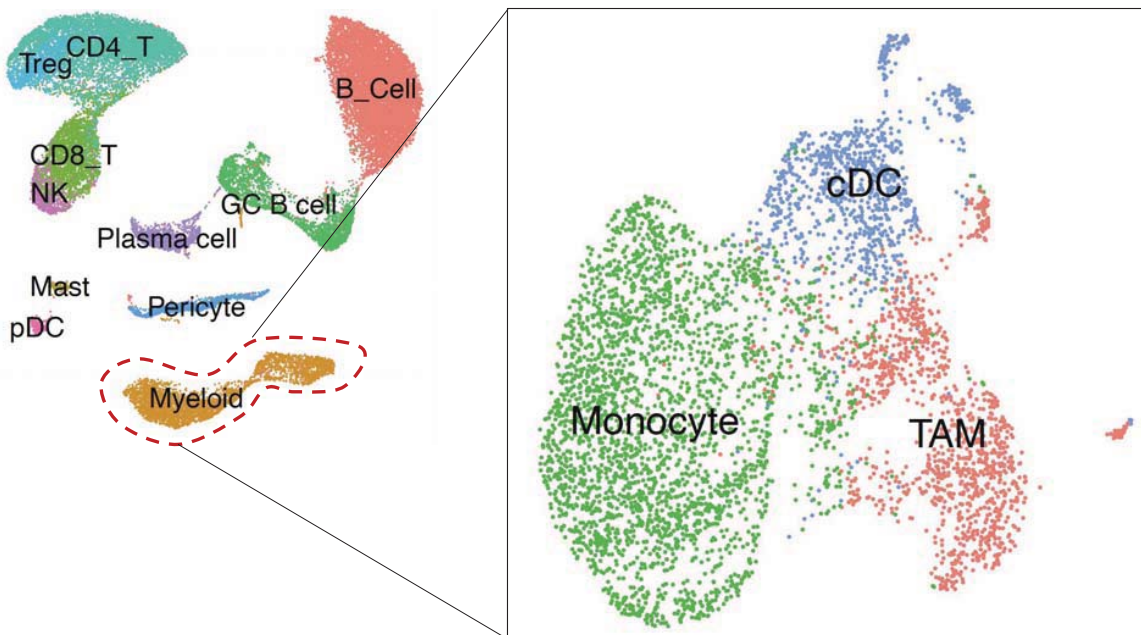
Why care
single cells?



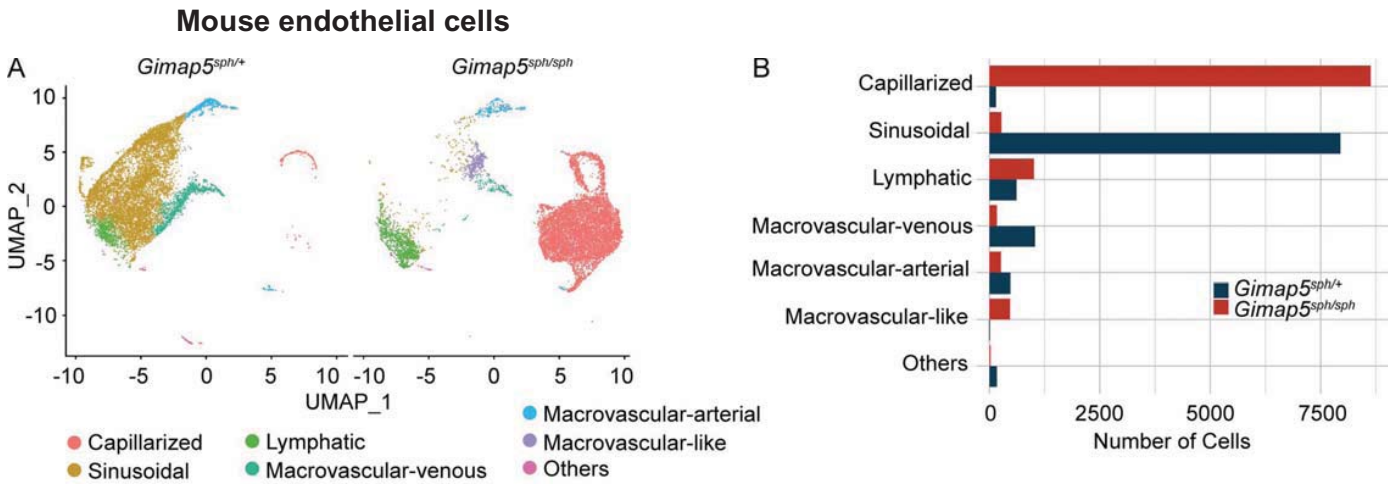
1. Single cells can identify and characterize cell types



Cell clusters of interest can be even further explored



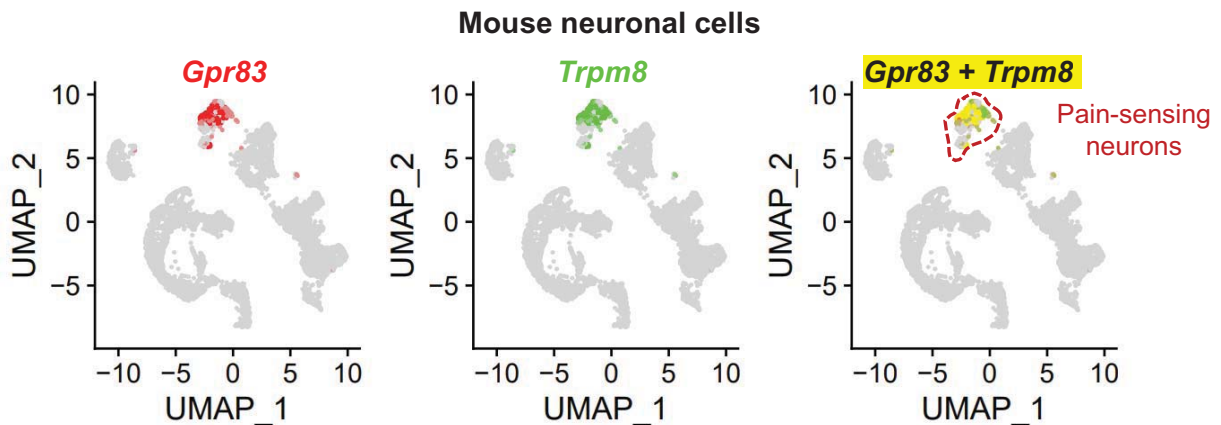
2. Single cell data can delineate cellular composition and their dynamics of cell state



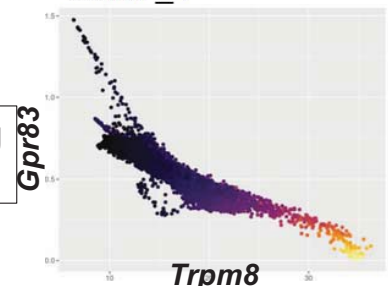
The homozygous deletion of *Gimap5* leads to the accumulation of a specific type of endothelial cells, resulting in portal hypertension

Drzewiecki & Choi *et al.*, 2021

3. Single cells can isolate strongly associated genes



Gpr83 and *Trpm8* interaction is observed in pain-sensing neurons exclusively, suggesting important roles



Kim *et al.*, 2022

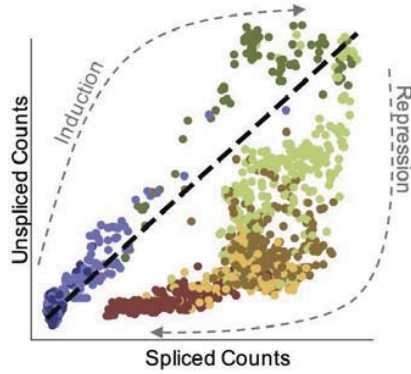
4. Single cell data can predict a continuum from static snapshots of cell state

A sequential trajectory of cliff diving



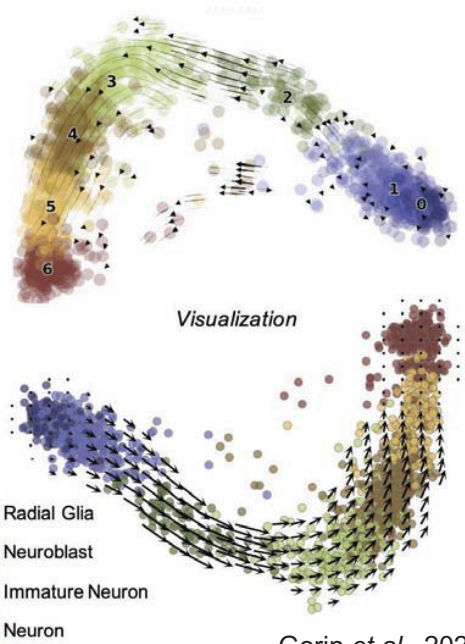
<https://images.app.goo.gl/DiAojKks7CLDdZwz6>

Neurogenesis



scVelo

velocyto

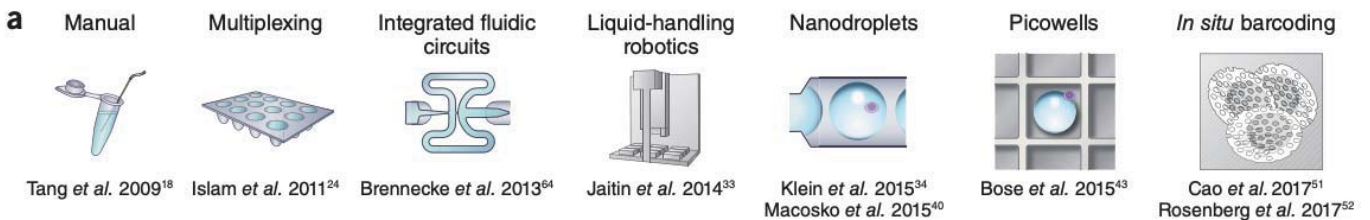


Visualization

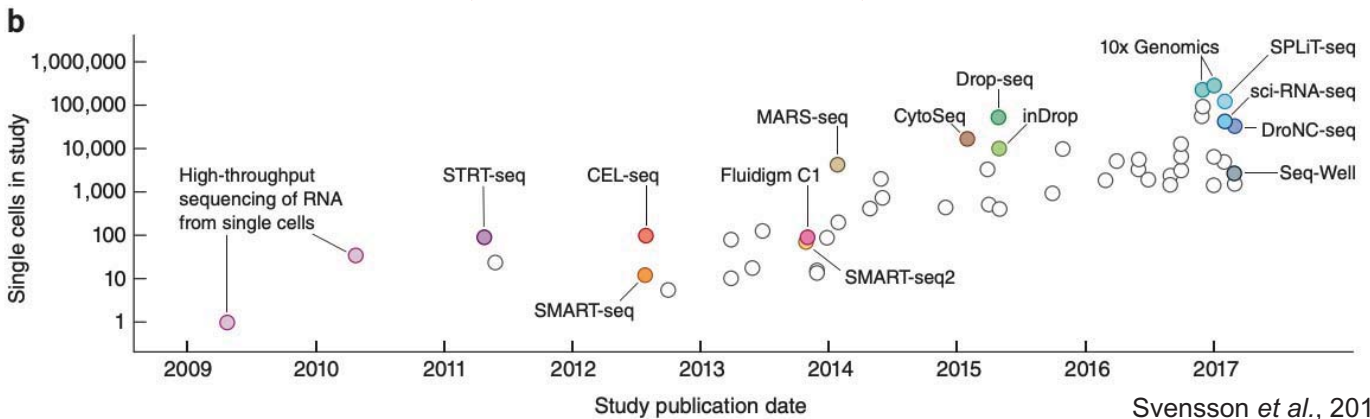
- Radial Glia
- Neuroblast
- Immature Neuron
- Neuron

Gorin et al., 2022

Exponential scaling of single cell sequencing tech.

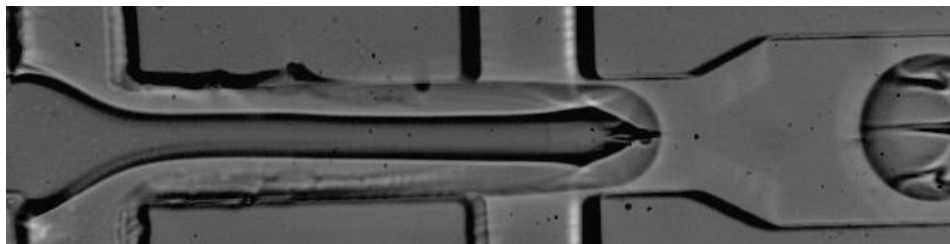
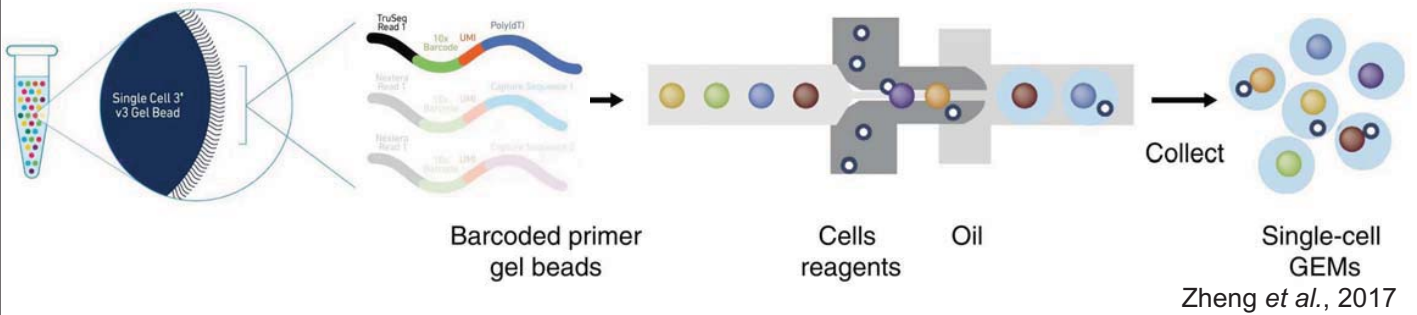


Few cells < 100s cells << 10,000 cells <<<<<< 100,000 cells



GAME-CHANGER

Microfluidic droplet-based cell isolation



A gif image from dropseq.org

Unfortunately, single cell transcriptomics loses context



The original brain

single-cell RNA Sequencing

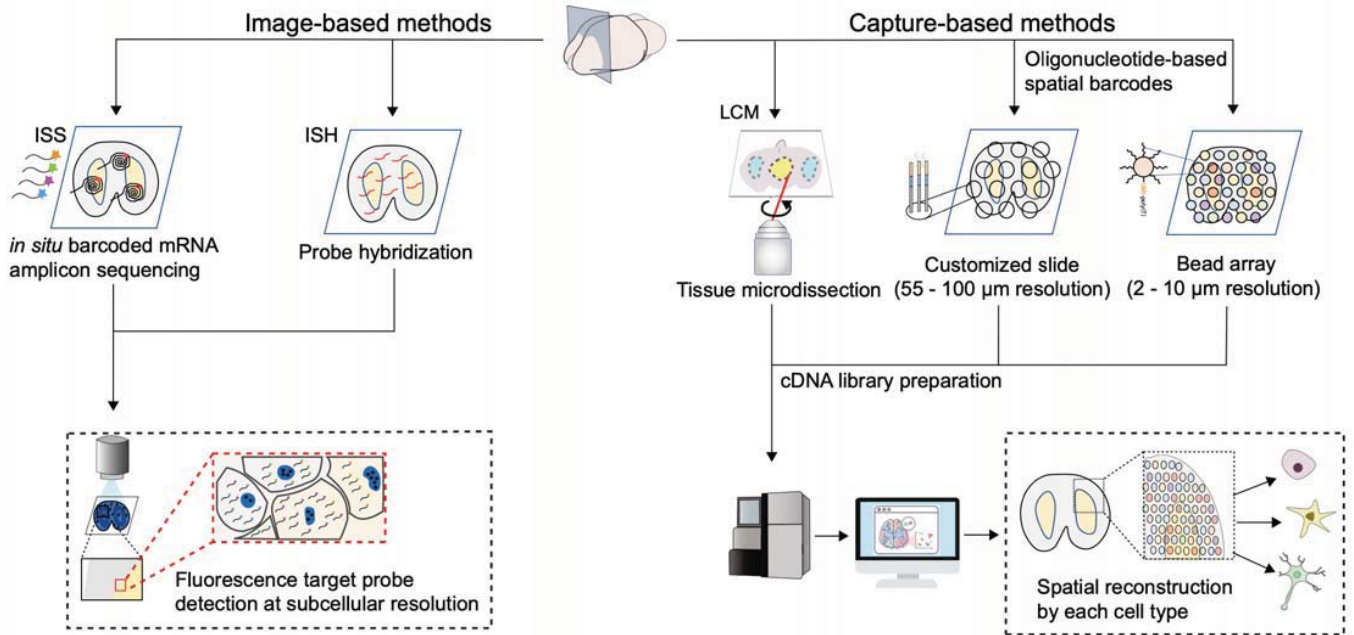
Spatially resolved transcriptomics

The first law of geography

*Everything is related to everything else,
but near things are more related than distant things*

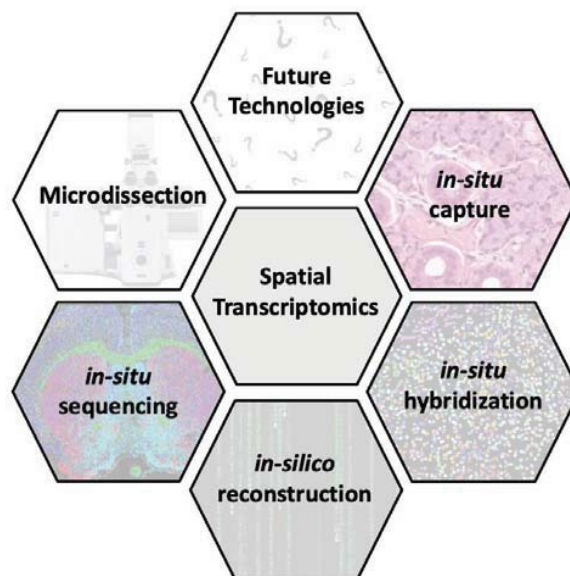
Image credit to Bo Xia

There are two types of SRT (SRT) methods



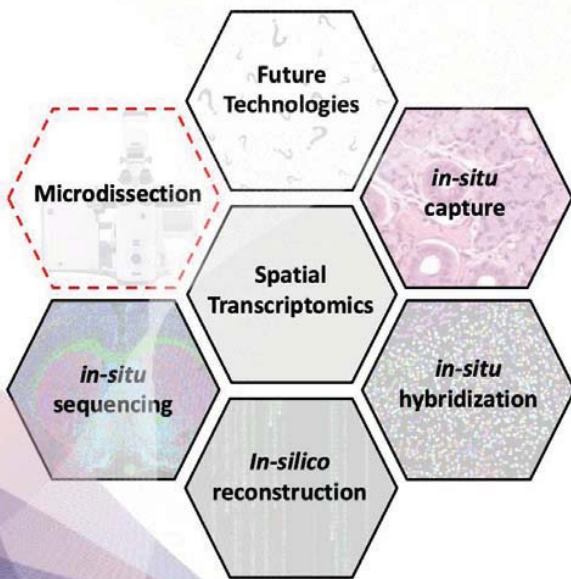
Lee et al., 2022

Various experimental spatially resolved techniques

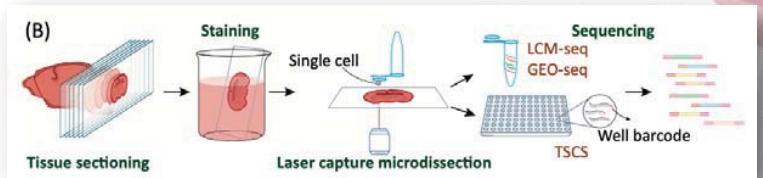


Asp et al., 2020

1. Microdissection-based technology



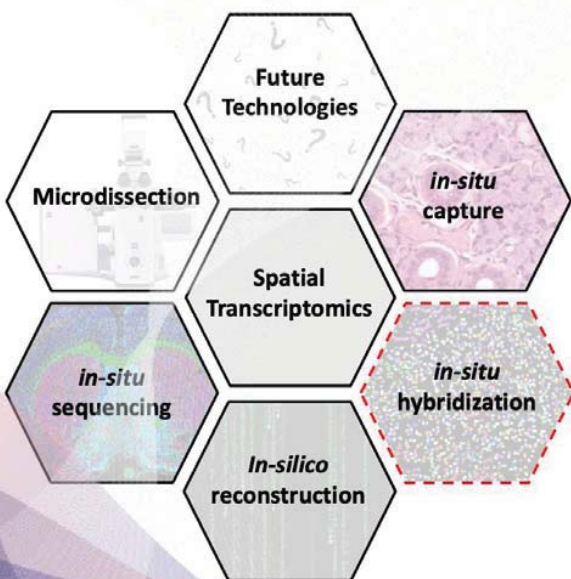
Asp et al., 2020



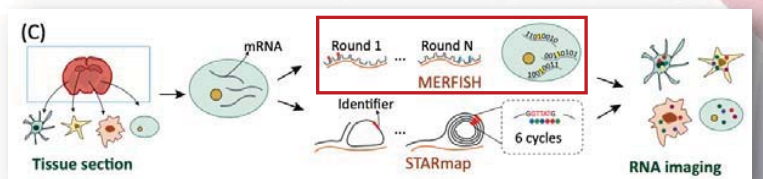
Liao et al., 2021

- Dissect a region of interest, place isolate in separate wells, and sequence
- Examples: Laser Capture Microdissection (LCM), Tomo-seq, TIVA, ProximID, and Niche-seq

2. *in situ* hybridization-based technology



Asp et al., 2020

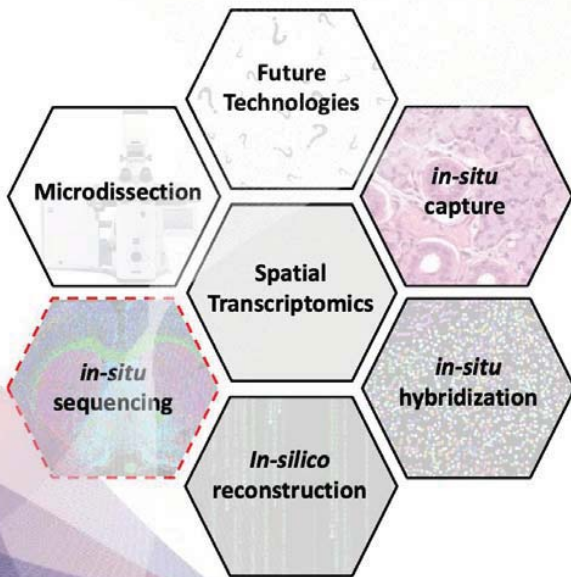


Liao et al., 2021

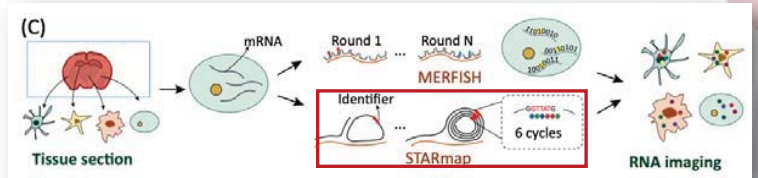
- Label probes for specific targets, hybridize in place
- Require “*a priori*” defined targets
- Examples) smFISH, seqFISH, **MERFISH**, seqFISH+, osmFISH, RNA Scope, DNA microscopy

* FISH – Fluorescence *In Situ* Hybridization

3. *in situ* sequencing-based technology



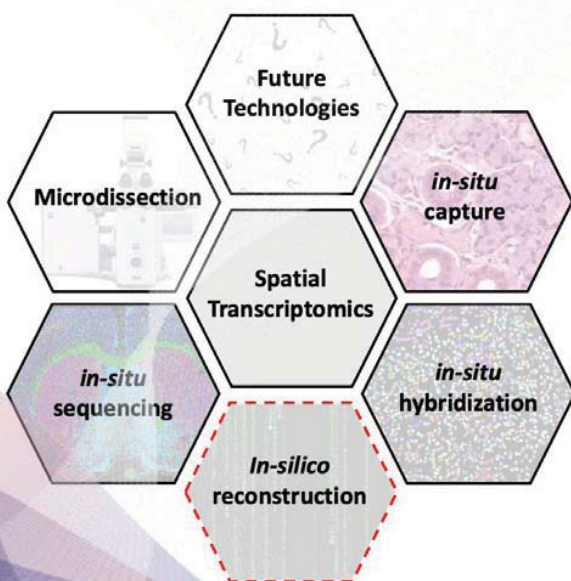
Asp et al., 2020



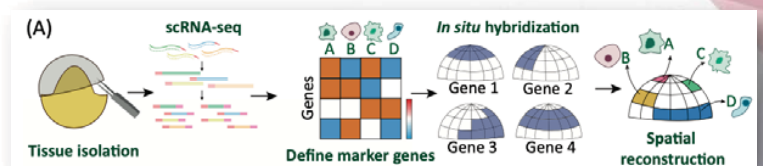
Liao et al., 2021

- Sequence the transcripts in place
- Offer sub-cellular resolution. Some rely on “*a priori*” defined targets
- Examples: ISS/CartaNa, BaristaSeq, **STARmap**, and FISSEQ

4. *in silico* reconstruction technology



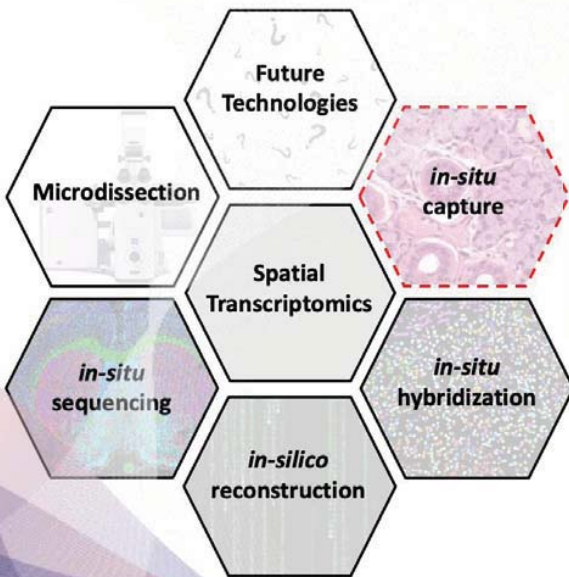
Asp et al., 2020



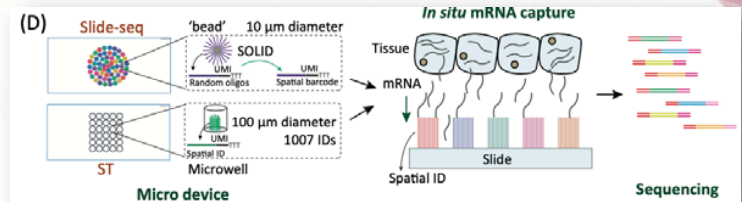
Liao et al., 2021

- Infer and reconstruct spatial structure from non-spatial data like single-cell transcriptomes
- At the core of this framework lies a structural correspondence hypothesis that cells in physical proximity share similar gene expression profiles
- Examples: novoSpaRc, CSOmap, and Seurat v3

5. *in situ* capture-based technologies



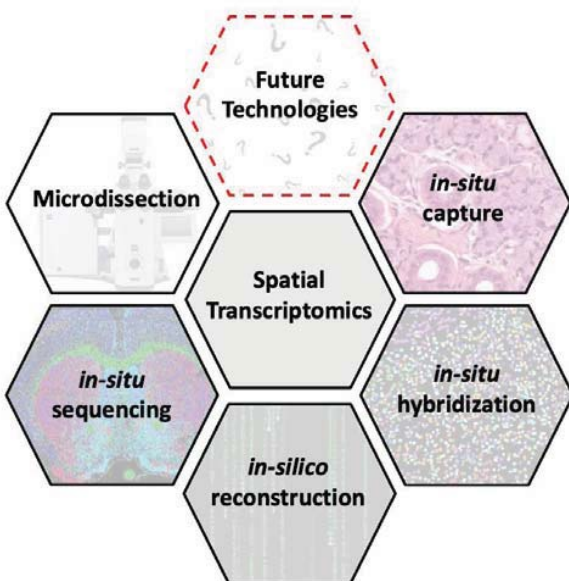
Asp et al., 2020



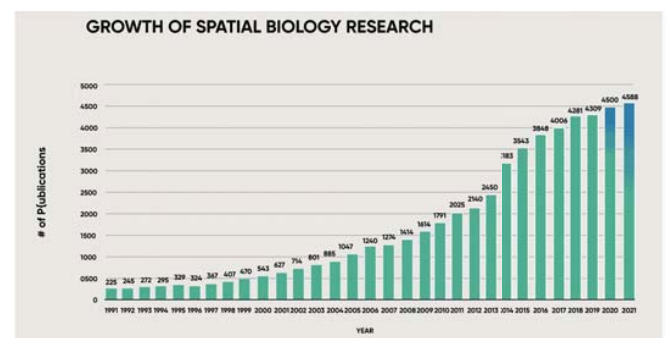
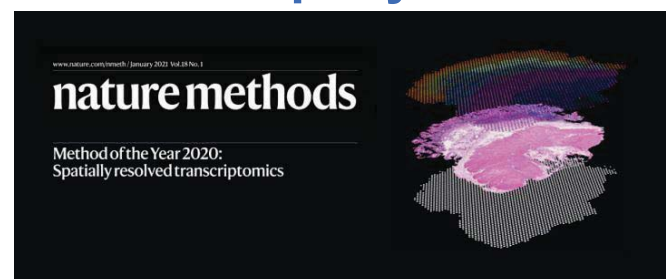
Liao et al., 2021

- Capture the transcripts *in situ* but sequence *ex situ*
- Usually less dependent on prior selection of targets
- Examples) Visium, **Spatial Transcriptomics (ST)**, **Slide-seq**, HDST, GeoMX, APEX-Seq, and Stereo-seq

SRT is an emerging class of high-throughput technologies and evolves rapidly



Asp et al., 2020

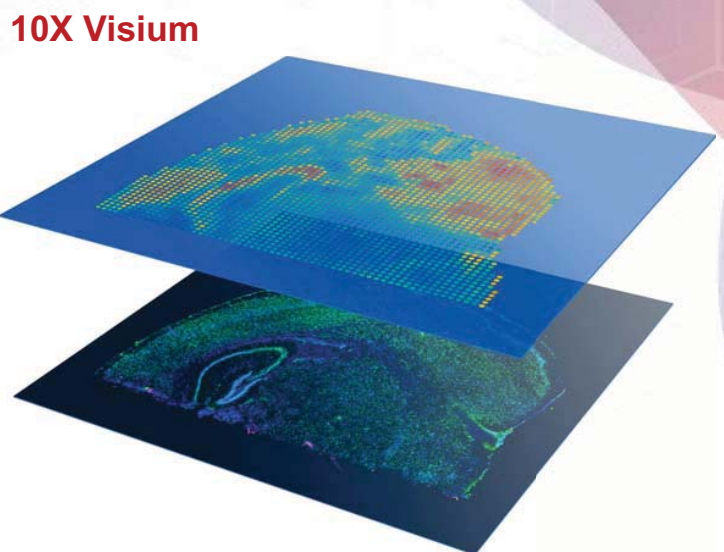
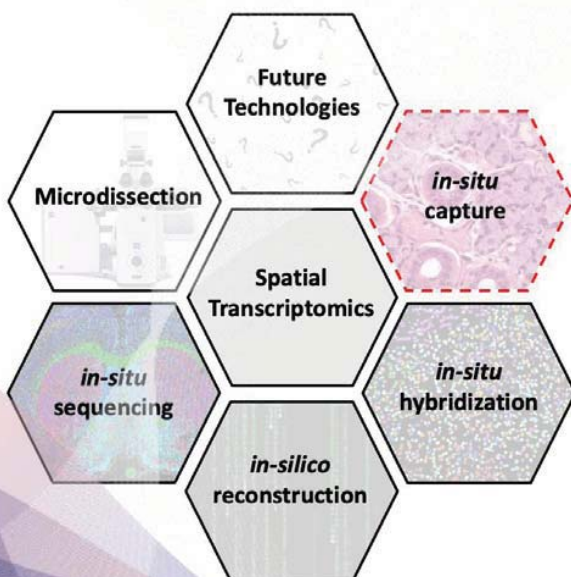


<https://images.app.goo.gl/Tpbyse6HWm7Lfdt8>

Recommended review literature on SRT

- Rao A, Barkley D, Franza GS, Yanai I. **Exploring tissue architecture using spatial transcriptomics**. Nature. 2021 Aug;596(7871):211-220. doi: 10.1038/s41586-021-03634-9. Epub 2021 Aug 11. PMID: 34381231; PMCID: PMC8475179.
- Longo SK, Guo MG, Ji AL, Khavari PA. **Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics**. Nat Rev Genet. 2021 Oct;22(10):627-644. doi: 10.1038/s41576-021-00370-8. Epub 2021 Jun 18. PMID: 34145435.
- Williams CG, Lee HJ, Asatsuma T, Vento-Tormo R, Haque A. **An introduction to spatial transcriptomics for biomedical research**. Genome Med. 2022 Jun 27;14(1):68. doi: 10.1186/s13073-022-01075-1. PMID: 35761361; PMCID: PMC9238181.
- Moses L, Pachter L. **Museum of spatial transcriptomics**. Nat Methods. 2022 May;19(5):534-546. doi: 10.1038/s41592-022-01409-2. Epub 2022 Mar 10. Erratum in: Nat Methods. 2022 Apr 19;: PMID: 35273392.
- Lee J, Yoo M, Choi J. **Recent advances in spatially resolved transcriptomics: challenges and opportunities**. BMB Rep. 2022 Mar;55(3):113-124. doi: 10.5483/BMBRep.2022.55.3.014. PMID: 35168703; PMCID: PMC8972138.

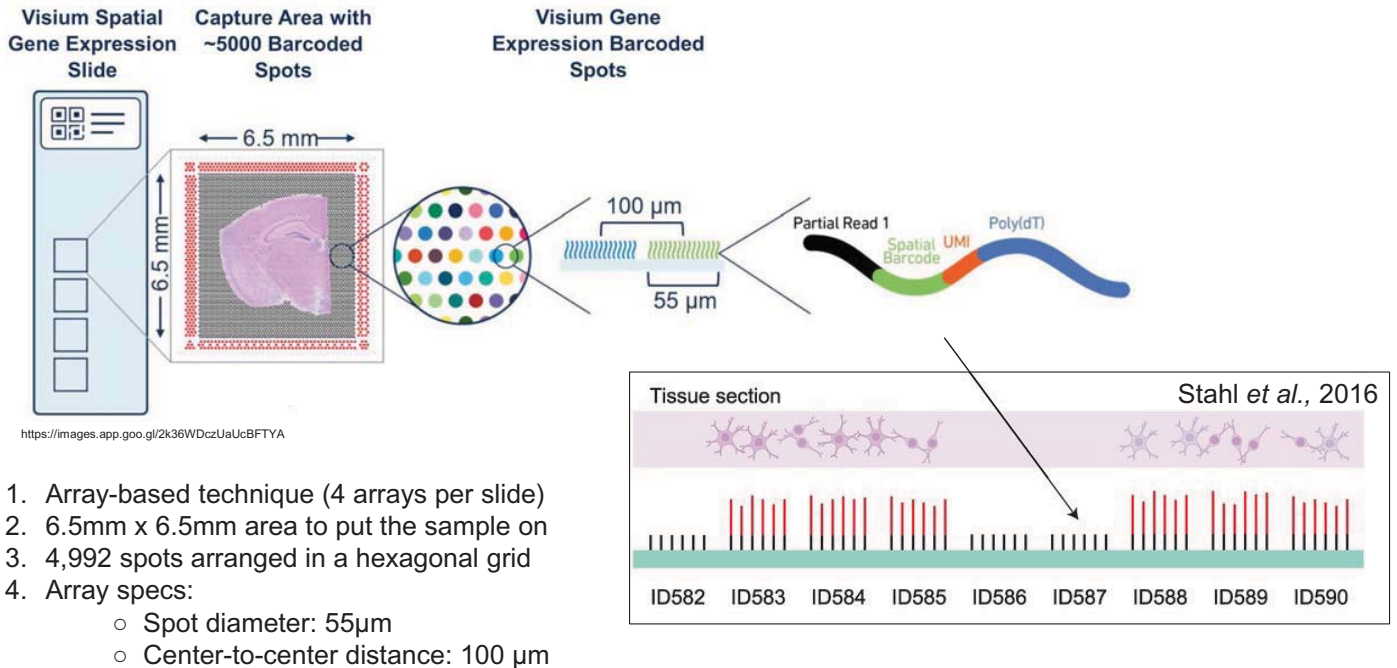
More in detail for *in situ* capture-based SRT



Asp et al., 2020

<https://images.app.goo.gl/VZFTmDkijQ38Vzm8>

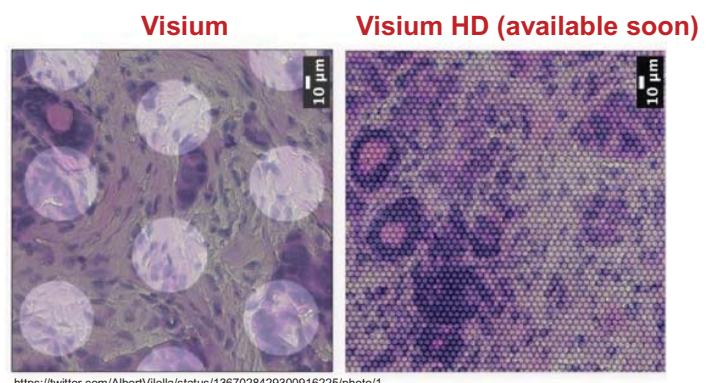
Capture-based SRT pertains spatial context :: Visium



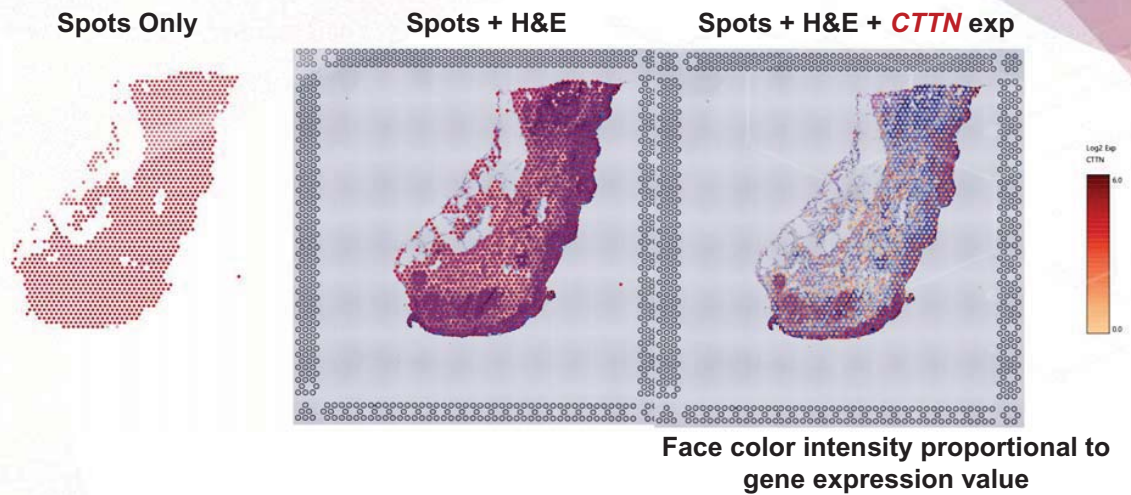
We are witnessing an evolution of Visium last decade



- Successor to Spatial Transcriptomics (ST)
- Approx. 1-10 cells contribute to each spot
Not a single-cell resolution!
- Data represented as [spot] x [gene] matrix
- You also get HE images of the same tissue



Human esophageal cancer 10X Visium data example



Face color intensity proportional to gene expression value

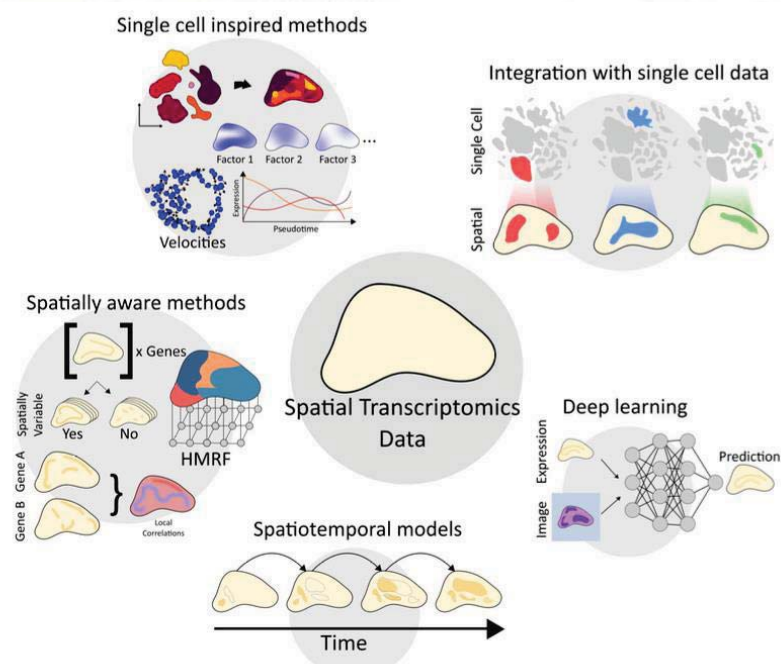
Will be performing hands-on exercise today

Computational data analysis methodology
for SRT

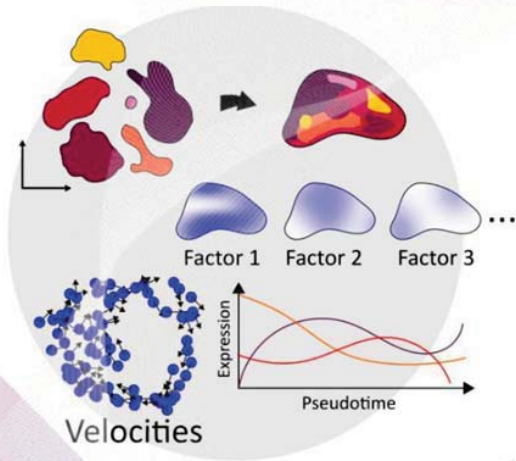
Generalized toolkits for spatial analysis

- R-based tools
 - ✓ **Seurat**
 - ✓ **STUtility** (extended spatial function for Seurat)
 - ✓ **Giotto** (greater variety of built-in tools for spatial analysis)
 - ✓ **SpatialExperiment**
- Python-based tools
 - ✓ **scanpy**
 - ✓ **squidpy** (extended spatial functions for scanpy)
 - ✓ **stLearn** (integrates spatial distance, tissue morphology and gene expression from spatial data)

Diverse technologies for SRT



Single cell inspired methods



Apply existing algorithms developed for single-cell data on spatial data

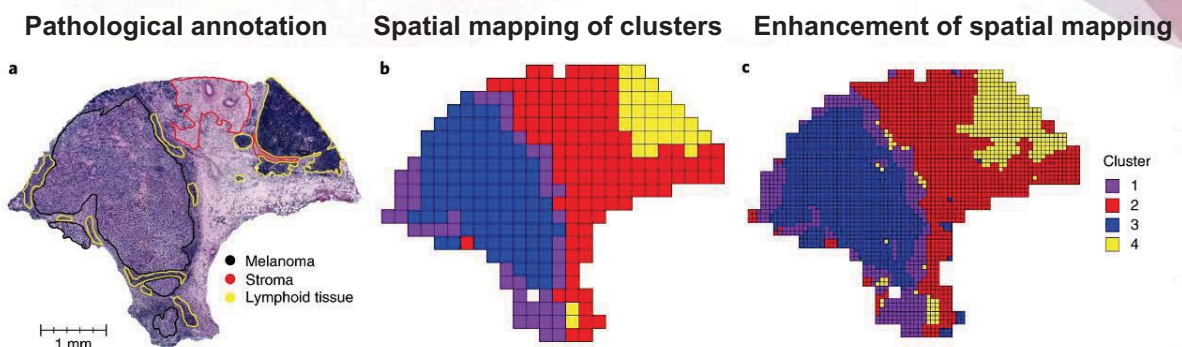
Example:

- Cluster spatial data and show clusters in space
- Factor models for data decomposition
- Trajectory inference

Available algorithms:

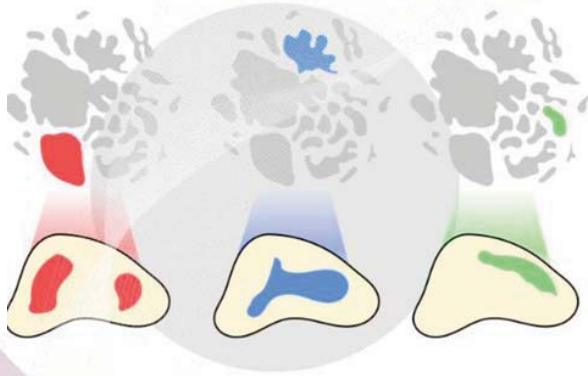
Seurat
scanpy
STUtility
stLearn
BayesSpace
SpatialExperiment (similar to SingleCellExperiment) etc.

Single cell inspired methods :: BayesSpace



Zhao *et al.*, 2021

Integration with single cell data



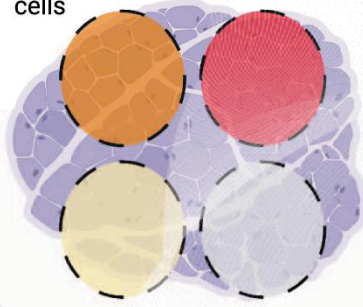
Use single-cell data as a reference when working with spatial data

Why integrate spatial data with single cells?

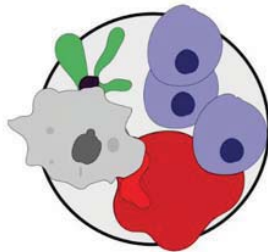
1. Efficient use of resources: Leverage extensive annotation already done for single-cell data
2. The problem of low resolution: Mixed cells in Visium spots

Integration with single cell data: mixed cell population

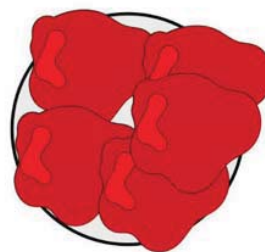
Each spot collects expression among several cells



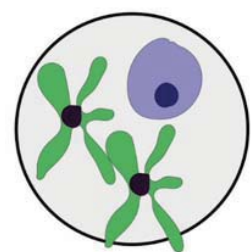
Spot 1



Spot 2

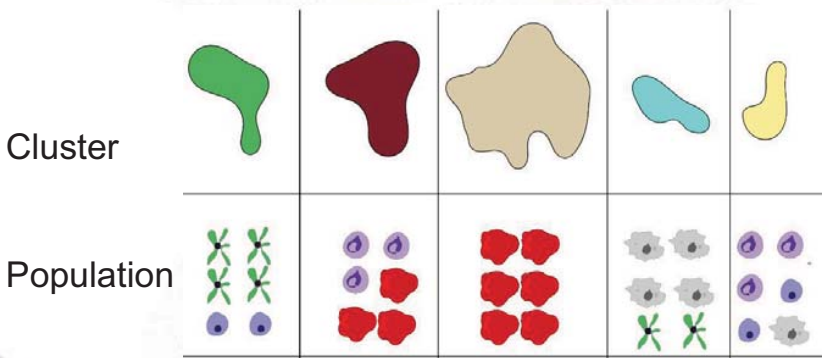


Spot 3



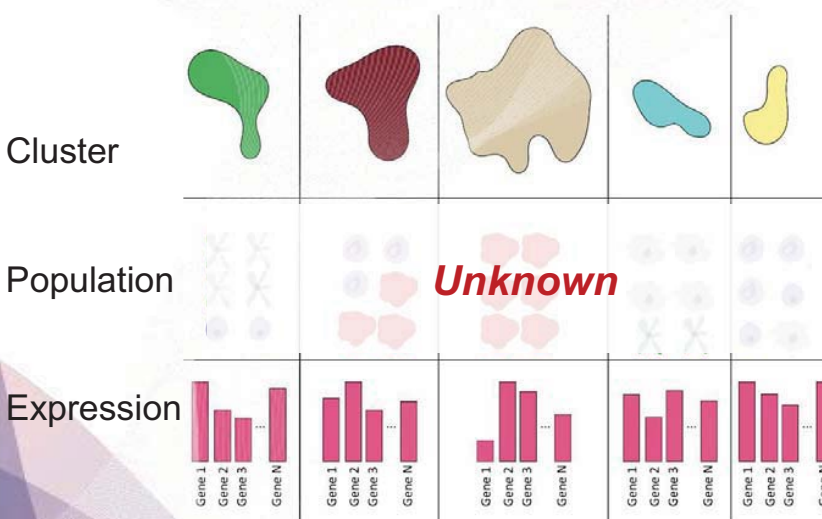
In several capture-based techniques (e.g., Visium and Slide-seq), observed expression values are a combination of multiple cell types and not all necessarily the same cell type.

Clusters do not represent single-cell type



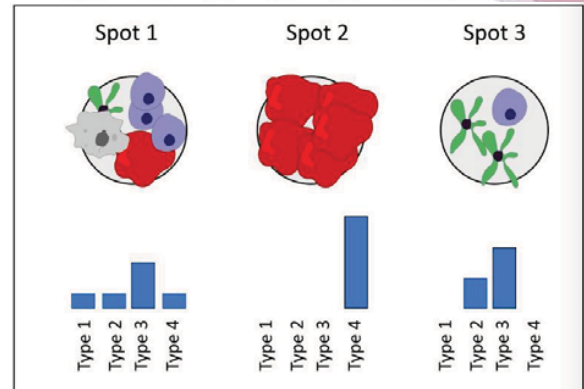
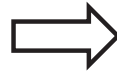
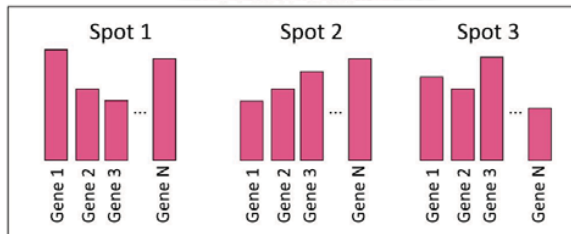
- Rather, clusters are more an assembly of spots with a similar composition of cell types
- We do not know what the cell type population looks like

We do not know what the cell type population look like



- Clusters do not represent single-cell type
- Rather, clusters are more an assembly of spots with a similar composition of cell types
- We only have observed expression

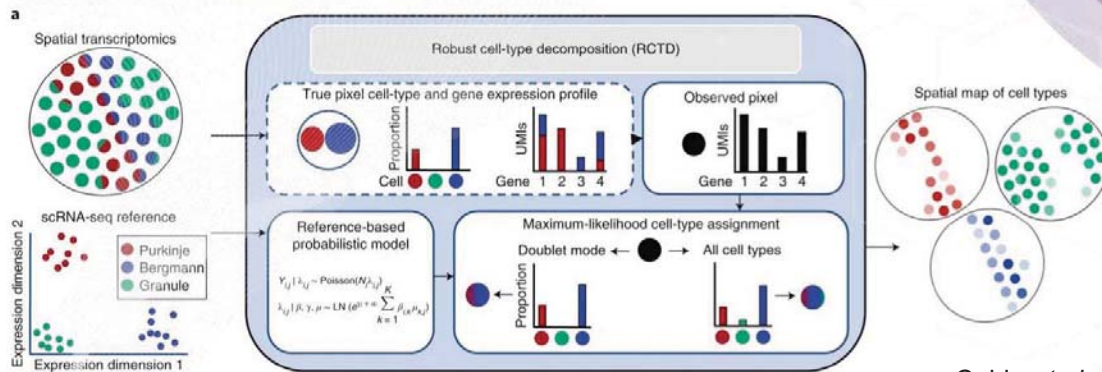
Estimate cell type abundance from observed gene expression profile



Different methods for integration with single cell data

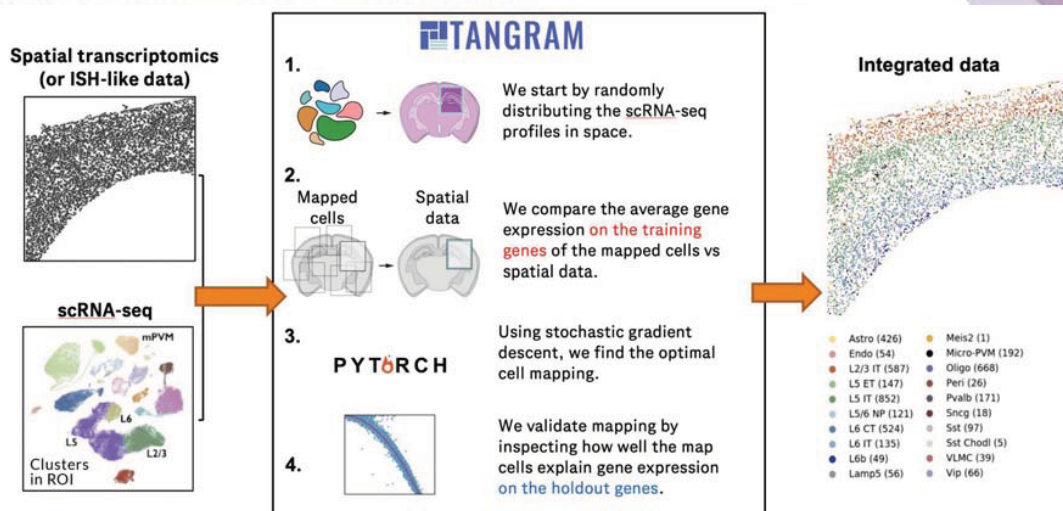
Marker gene	Anchor	Probabilistic Model	Optimization
Extract marker genes for each cell type from single cell data	Find anchors between single cell and spatial data. Create correction vectors based on expression differences	Assume gene expression follows a certain statistical distribution. Joint model for single cell and spatial data.	Find spatial location where each cell most likely to reside
Compute enrichment score for each set of marker genes in spatial locations	Use correction vectors to integrate two data sets. Transfer labels of single cells to spatial data	Learn cell type parameters from single cell data and use them to deconvolve spatial data	Simultaneously optimize terms such as: <ul style="list-style-type: none"> • Cell density • UMI distribution • Gene distribution
Moncada <i>et al.</i> , 2020	Seurat	Stereoscope, RCTD, cell2location	Tangram

Integration with single cell data :: RCTD



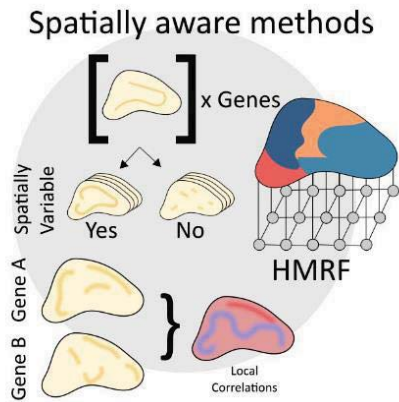
- A probabilistic model for inferring cell types in spatial transcriptomics data, supervised with a labeled single-cell RNA-seq reference
- Robust cell-type decomposition (RCTD) uses **maximum likelihood estimation** to identify cell types present on each spatial transcriptomics spot, in addition to estimating cell type proportions
- Robust decomposition of cell type mixture in spatial transcriptomics

Integration with single cell data :: Tangram



Via integration, *Tangram* creates new spatial data by aligning the scRNAseq profiles in space. This allows to project every annotation in the scRNAseq (e.g. cell types, program usage) on space.

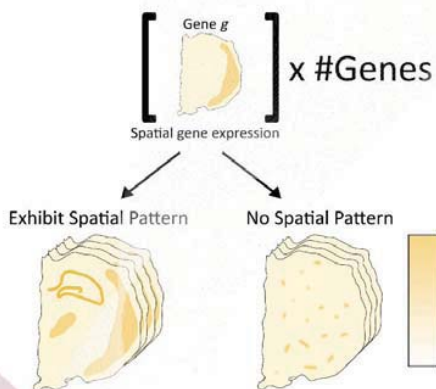
Spatially aware methods



Attempts to include knowledge of spatial structure in the analysis, not only to visualize results

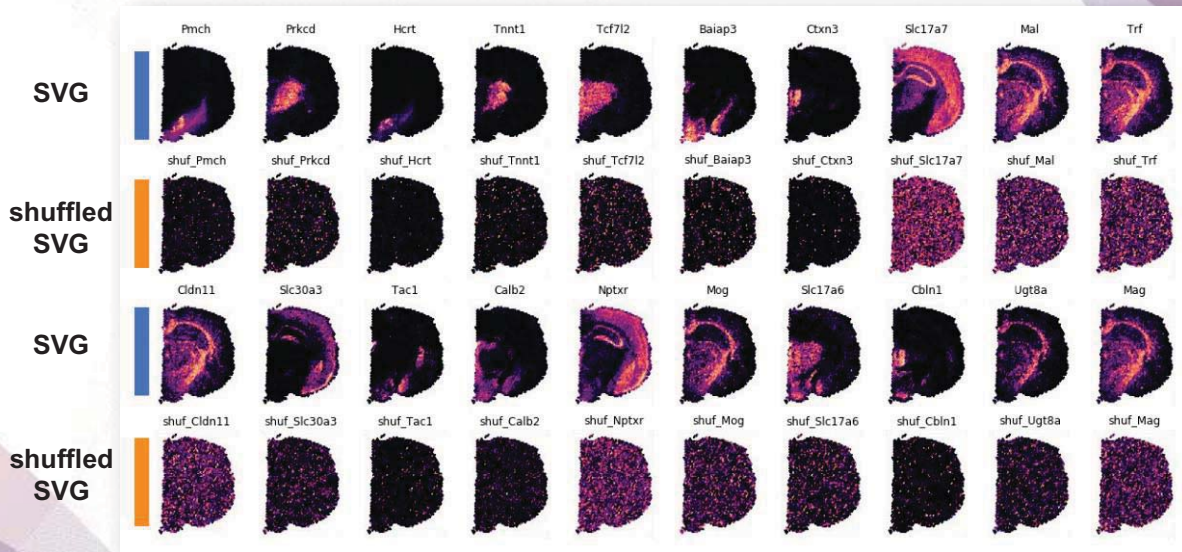
- Identifying spatially variable genes and features
- Finding spatially coherent expression domains
- Leveraging spatial proximity to increase the robustness of inference
- Finding local correlations between features

Spatially aware methods :: spatially variable genes



- Spatially variable genes (SVGs) are genes with a highly spatially correlated pattern of expression, which varies along with the spatial distribution of a tissue structure of interest
- Standard statistical measures such as Moran's I or Geary's C can be used to rank genes
- SpatialIDE, SVCA, and SPARK use probabilistic models
- Essentially, test whether a "spatial" term in the covariance function significantly increases model's ability to explain data

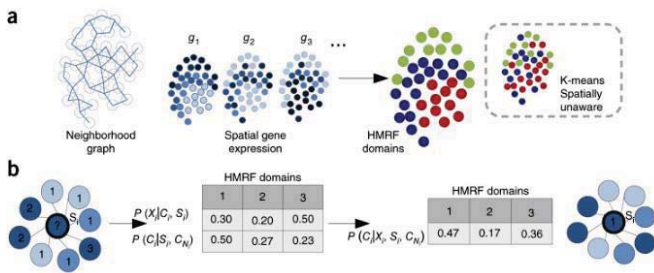
Spatially aware methods :: spatially variable genes



- 20 gene expression profiles from mouse brain
- Shuffle spots to get random expression profiles (with “shuf” prefix)

Spatially aware methods :: spatial domain patterns

HMMRF (Hidden Markov Random Field)

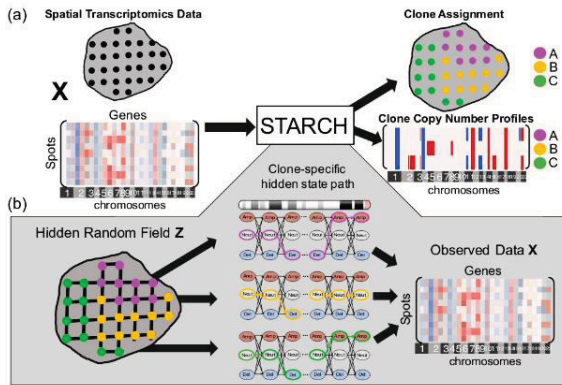


Zhu *et al.*, 2018

gene expression (x_i) and neighborhood configuration (c_{iN})

- Normal clustering mainly focus on gene expression
- Leverage spatial information to find spatially coherent clusters (domains)
- Normally use **HMMRF**
- Construct a graph based on spatial proximity
- The probability of a node (spot) belonging to a specific domain depends on:
 - Agreement with a domain expression profile
 - Coherence with neighbors

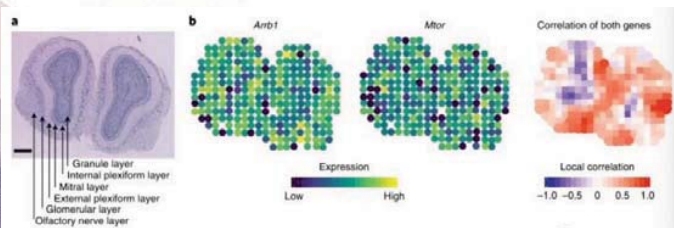
Spatially aware methods :: spatial domain patterns



Elyanow *et al.*, 2021

STARCH

- Infers Copy Number Aberrations (CNA) from spatial transcriptomics data
- Increases robustness of inference by aggregating data in the same domains (similar profiles)
- It uses Hidden Markov Random Fields (HMRF)

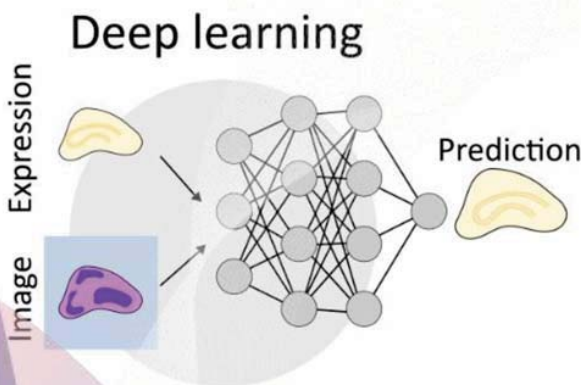


Ghazanfar *et al.*, 2020

scHOT

- Computes (spatially) weighted correlations to find local correlations
- It also uses Hidden Markov Random Fields (HMRF)

Deep learning



Apply deep learning to spatial data (very broad)

- Relatively few examples. Limited amount of high-quality available data.

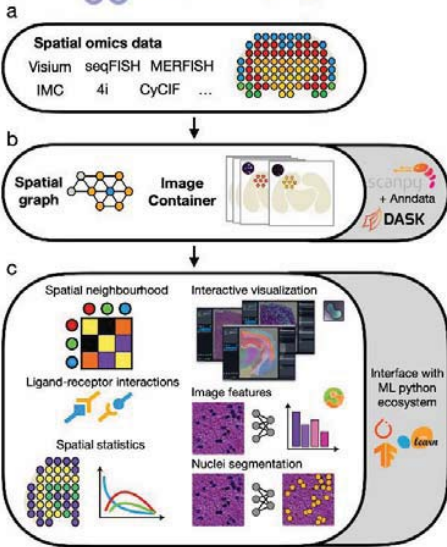
Current examples:

- **XFuse**: “superresolution” (pixel) of gene expression by learning joint representation of image and expression data.
- **stPlus**: Uses scRNA-seq data and autoencoders to enhance spatial transcriptomics data
- **SpaGCN**: simultaneous domain and SVG detection using graph convolution layers
- **RESEPT**: Uses graph convolutional network to embed spatial data in RGB space, then uses a CNN to segment data into spatially coherent tissue domains

Computational suites: squidpy



4

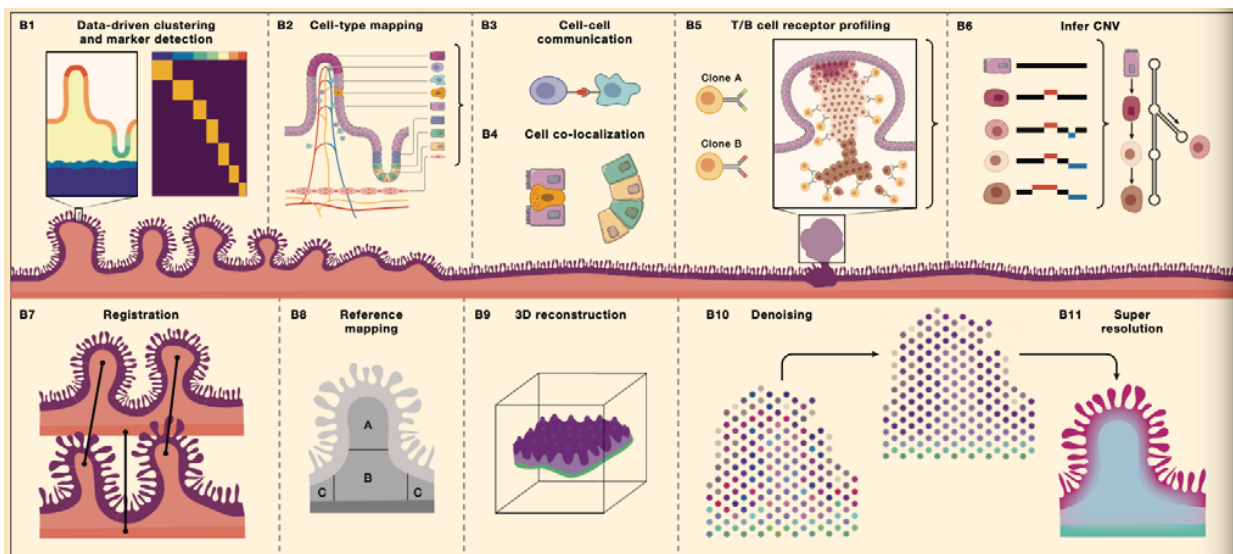


“One framework to rule them all, one framework to find them...”

- It builds on top of scanpy and anndata, from which it inherits modularity and scalability.
- Tailored towards spatial data with support for multiple different experimental platforms (not only Visium)
- Easy to construct spatial graphs and perform graph operations
- Has excellent interface with ML ecosystems such as PyTorch, TensorFlow and sklearn

Palla *et al.*, 2022

Snapshots of spatial transcriptomics applications



Larsson *et al.*, 2023

Take home messages

- ✓ There are tons of spatial techniques
- ✓ An ever-increasing repertoire of computational methods!
 - A lot of tools out there, but sometimes beneficial to construct custom solutions
- ✓ Spatial-omics data is already improving our understanding of human health and disease in research, diagnostic, and therapeutic setting

KSBi-BIML 2023

Analysis of Spatially resolved transcriptomics

Welcome to Spatial Transcriptomic session

R/Python script, objects, power point slides and can be downloaded here:
https://www.dropbox.com/sh/v4k9hwwlwhab8pz/AAC3ta-o_LdawSGQ_--2UWdDa?dl=0

Steps	File name	Input/Output
chapter 5 single cell reference	BIML_sc_allen_cortex.rds	chapter 5, 6 input
chapter 5 processed visium data	BIML_visium_brain_processed.rds	chapter 5 output
chapter 5 processed visium data	BIML_visium_cortex_processed.rds	chapter 5 output
chapter 6 spacexr doublet mode metadata	BIML_spacexr_cortex_doubletmode.csv	chapter 6 output
chapter 6 spacexr full mode assay	BIML_spacexr_cortex_fullmode.csv	chapter 6 output
chapter 6 spacexr/seurat meta 저장된 visium	BIML_visium_cortex_annotated.rds	chapter 7, 8 input
chapter 7 conda environment info	environment.yml	chapter 7 python environment
chapter 7 anndata	BIML_visium_cortex_anndata.h5ad	chapter 7 input/output
chapter 7 jupyter notebook	BIML_chapter7_squidpy_code.ipynb	chapter 7 python script
chapter 8 visium json	BIML_scalefactors_json.json	chapter 8 input
chapter 8 cellchat prop calculated object	BIML_visium_cortex_prob_cellchat.rds	chapter 8 output
chapter 9 stutility input folder	BIML_visium_STutility_input_files	chapter 9 input
chapter 9 stutility object	BIML_visium_cortex_stutility.rds	chapter 9 output

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

Table of Contents

Chapter 1	What is R programming?	
Chapter 2	Then why python?	
Chapter 3	What is single cell RNA-seq and spatial transcriptomics?	
Chapter 4	What is Space Ranger?	Exercise
Chapter 5	Integrative analysis of spatial datasets - Seurat	
Chapter 6	Decomposition/Mapping analysis – Seurat/SpaceXR	
Chapter 7	Neighborhood analysis and co-occurrence - Squidpy	
Chapter 8	Cell-cell interaction - CellChat	
Chapter 9	Visualization of blended spatial plot of several features – STutility	
Chapter 10	Summary	
Chapter 11	Q&A	

3

1. What is R programming?

What is R and Why R?

- R is used widely in biological research and provides a solid platform for beginner scientific programmers.
- It's free and open-source.
- Bioconductor uses the R statistical programming language and is open source and open development.



Modified from : <https://datacarpentry.org/semester-biology/about/why-r/>
<https://www.bioconductor.org/>

Several ways to use R

Command line Interface

```
(base) harim ~$ R
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Graphical user interface

Rstudio

```
code.R
1 x <- 'Hello, World!'
2 print(x)

Environment History Connections
R - Global Environment
Values
x "Hello, World!"

2.9 (Top Level)
R Script
Console Terminal Jobs
~/
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/en_US.UTF-8
TF-8/en_US.UTF-8

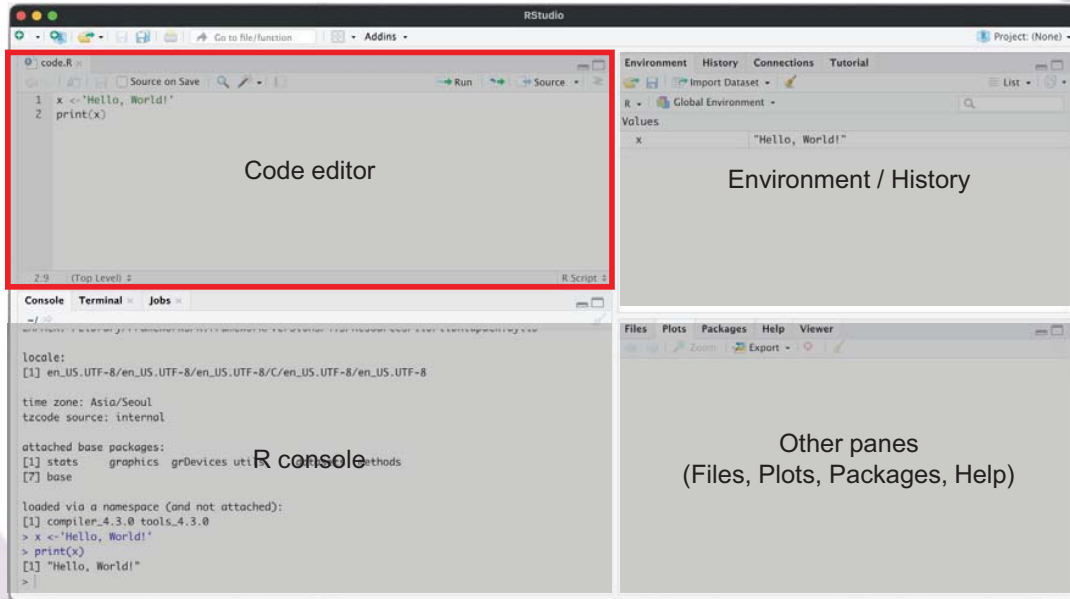
time zone: Asia/Seoul
tzcode source: internal

attached base packages:
[1] stats graphics grDevices utils datasets
ets methods
[7] base

loaded via a namespace (and not attached):
[1] compiler_4.3.0 tools_4.3.0
> x <- 'Hello, World!'
> print(x)
[1] "Hello, World!"
>
```

Rstudio

- Rstudio – Integrated development environment (IDE) for R and python



Code editor에서 코드를 작성하거나 적혀져 있는 코드를 실행합니다.
Ctrl+Enter를 통해 각 한 줄씩의 코드를 실행할 수 있습니다.

7

2. Then why python?

Python provides analysis tools that R does not

- Some analysis tools are solely based on python



- Here we will use squidpy for neighborhood enrichment analysis, which R does not provide, on **chapter 6**



CellphoneDB

9

3. What is single cell RNA-seq and spatial transcriptomics?

Single-cell transcriptomics

Examines the gene expression level of individual cells in a given population by simultaneously measuring the RNA concentration (conventionally only messenger RNA (mRNA)) of hundreds to thousands of genes.

<https://www.10xgenomics.com/single-cell-technology>

Spatial transcriptomics

Since Visium Spatial Gene Expression is a spatial transcriptomics solution, you can analyze the transcriptome within the tissue context.

Visium Spatial Gene Expression works with cell capture slides that contain four capture areas with 5,000 barcoded spots. These barcoded spots include capture oligonucleotides that bind to the RNA in the tissue.

<https://www.10xgenomics.com/spatial-transcriptomics>

4. What is Space Ranger

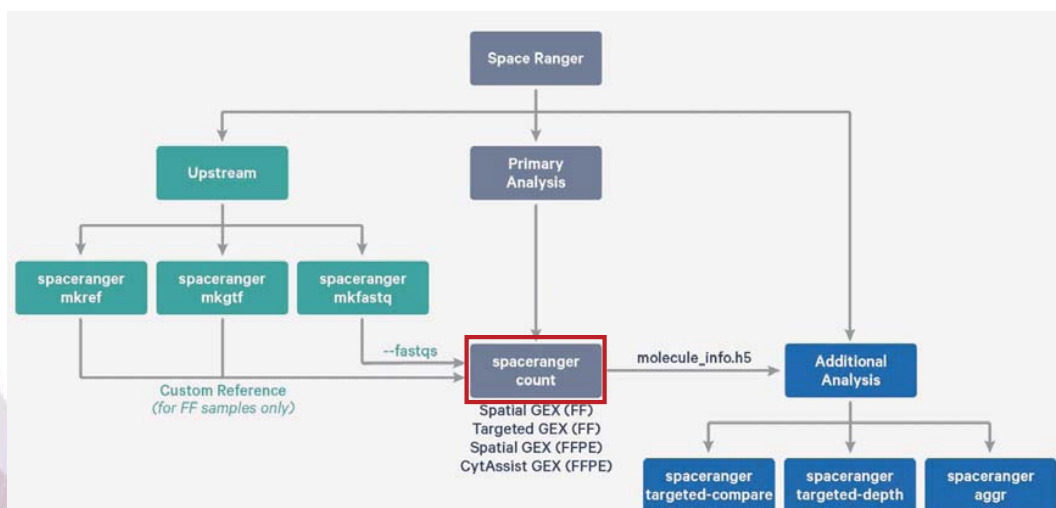
What is Space Ranger?

- Space Ranger is a set of analysis pipelines for processing 10X Genomics Visium sequence data (FASTQ files) with high resolution microscope images of tissue.
- It maps the transcriptomic reads to the microscope image of the tissue
- We will introduce **spaceranger count** pipeline

13

Space Ranger Pipelines

- spaceranger mkfastq
- **spaceranger count**
- spaceranger aggr
- spaceranger targeted-compare
- spaceranger targeted-depth



<https://support.10xgenomics.com/spatial-gene-expression/software/pipelines/latest/what-is-space-ranger>

14

Installing Space Ranger

1. Download and unpack the Space Ranger .tar.gz file in any location

Terminal

```
$cd /opt
$tar -xzvf spaceranger-2.0.0.tar.gz
```

2. Download and unpack proper reference data .tar.gz file in a convenient location

```
$tar -xzvf refdata-gex-GRCh38-2020-A.tar.gz
```

3. Pre-pend the Space Ranger directory to your \$PATH

```
$export PATH=/opt/spaceranger-2.0.0:$PATH
```

<https://support.10xgenomics.com/spatial-gene-expression/software/pipelines/latest/installation>

15

Run *spaceranger* count command

Terminal

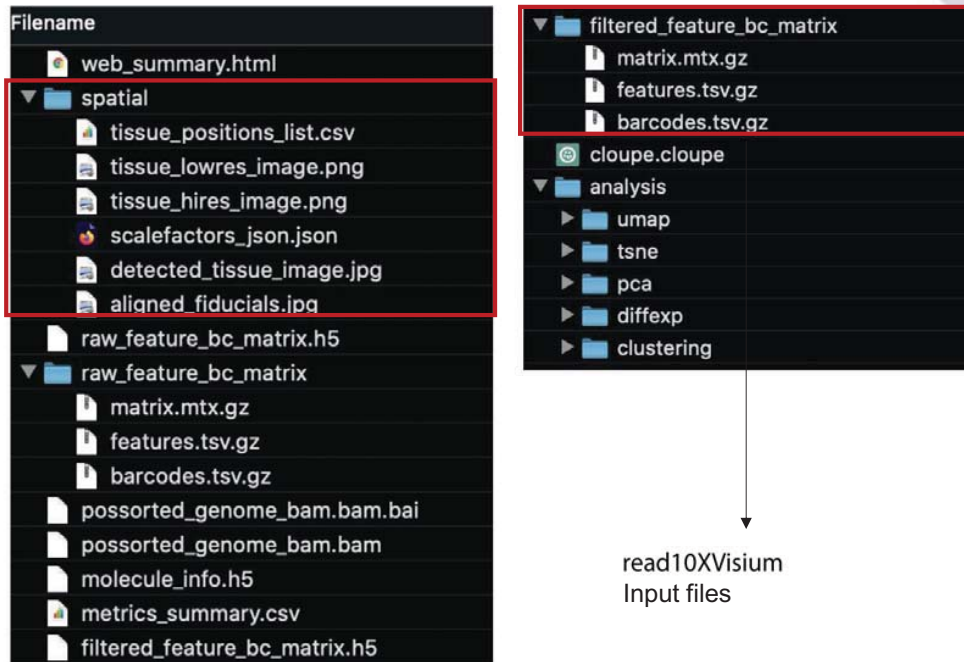
```
$cd /home/jdoe/runs
$spaceranger count --id=sample345 \ #Output directory
                   --transcriptome=/home/jdoe/refdata/GRCh38-2020-A \ #Path to Reference
                   --fastqs=/home/jdoe/runs/HAWT7ADXX/outs/fastq_path \ #Path to FASTQs
                   --sample=mysample \ #Sample name from FASTQ filename
                   --image=/home/jdoe/runs/images/sample345.tiff \ #Path to brightfield
image
                   --slide=V19J01-123 \ #Slide ID
                   --area=A1 \ #Capture area
                   --localcores=8 \ #Allowed cores in localmode
                   --localmem=64 #Allowed memory (GB) in localmode
```

<https://support.10xgenomics.com/spatial-gene-expression/software/pipelines/latest/using/count>

- Input : the microscope image (.tiff), FASTQ files(Fastq)
- Perform : sequence alignment, tissue detection
- Output : gene-spot matrix

16

Output files of Space Ranger



17

Output files

- **raw_feature_bc_matrix**
Dataset having spots that theoretically don't overlap with tissue
- **filtered_feature_bc_matrix**
Dataset filtered to the spots overlapping tissue, as determined by Loupe Browser (Visium) alignment file
- **tissue_positions_list.csv**
The spot coordinates information is stored
- **scalefactors_json.json**
Scaling factors that convert spot coordinates to pixel coordinates
- **metrics_summary.csv**
Metrics displayed in the interactive website
- **web_summary.html**
Interactive website

18

Web summary .html file

DLPFC_Br8492_post_manual_alignment

Summary
Analysis

4,618

Number of Spots Under Tissue


48,988

Mean Reads per Spot

1,137

Median Genes per Spot

Spots ?



Sequencing ?

Number of Reads	226,227,723
Valid Barcodes	96.8%
Valid UMIs	100.0%
Sequencing Saturation	95.2%
Q30 Bases in Barcode	94.0%
Q30 Bases in RNA Read	93.3%
Q30 Bases in UMI	94.4%

<https://lmweber.org/OSTA-book/space-ranger-visium.html>

Web summary .html file

Metrics	Definition	Expected/Recommended value
Number of Spots Under Tissue	The number of barcodes associated with a spot under tissue.	
Mean Reads per Spot	The number of reads, both under and outside of tissue, divided by the number of barcodes associated with a spot under tissue.	Recommended: 50,000
Median Genes per Spot	The median number of genes detected per spot under tissue-associated barcode. Detection is defined as the presence of at least 1 UMI count.	Vary by sample, and low values are not necessarily indicative of a failed experiment
Number of reads	Total number of read pairs that were assigned to this library in demultiplexing.	Fresh frozen libraries: a minimum of 50k FFPE v1 libraries: a minimum of 25k
Valid Barcodes	Fraction of reads with barcodes that match the whitelist* after barcode correction	Expected >75%
Valid UMIs	Fraction of reads with valid UMIs	Expected >75%
Sequencing Saturation	The fraction of reads originating from an already-observed UMI	Dependent upon sequencing depth and sample complexity (at last 60-80 % in most applications)
Q30 bases in barcode, Sample Index, or UMI	Fraction of tissue-associated barcode, Sample Index, or UMI bases with Q-Score >= 30, excluding very low quality/no call (Q<2) bases from the denominator	Sequencing platform dependent (Most Illumina runs generate >70-80% Q30 data)

<https://lmweber.org/OSTA-book/space-ranger-visium.html>

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
- Generally, 1~2 GB each

21

Loupe Browser

10x Genomics

Spot Opacity

Spatial

Categories

Graph-Based

- Cluster 1 (1233)
- Cluster 2 (990)
- Cluster 3 (552)
- Cluster 4 (526)
- Cluster 5 (420)
- Cluster 6 (190)

Manual Annotation of spots

Significant Feature Comparison

Globally Distinguishing

Feature Type

Gene

Graph-Based: Up-Regulated Genes Per Cluster

Name	Cluster 1	P-Value	Cluster 2	Cluster 3	Cluster 4
DIRAS2	0.89 ***	5.86e-3	-0.60	-0.53	-0.24
MAP2K1	0.89 ***	6.91e-3	-0.64	-0.78	-0.26

<https://www.10xgenomics.com/spatial-gene-expression/software/visualization/latest/analysis>

22

Download R packages to be used in analysis

```
install.packages("ggplot2")
install.packages("devtools")
install.packages("remotes")
install.packages("dplyr")
install.packages("anndata")
install.packages("cowplot")
install.packages('Seurat')
devtools::install_github('satijalab/seurat-data')
devtools::install_github("thomasp85/patchwork")
devtools::install_github("dmcable/spacexr", build_vignettes = FALSE)
remotes::install_github("jbergenstrahle/STUtility")
remotes::install_github("sqjin/CellChat")
```

Load packages

```
library(ggplot2)
library(devtools)
library(remotes)
library(dplyr)
library(anndata)
library(cowplot)
library(Seurat)
library(SeuratData)
library(patchwork)
library(spacexr)
library(STUtility)
library(cellchat)
# For reproducibility set the seed
set.seed(1234)
```

5. Integrative analysis of spatial datasets - Seurat

Dataset Description

We will be using a recently released dataset of sagittal mouse brain slices generated using the Visium v1 chemistry.

There are two serial anterior sections, and two (matched) serial posterior sections.

Biological annotations of spots (i.e., cell group information) are predicted using Seurat (https://satijalab.org/seurat/articles/spatial_vignette.html).

25

Load and preprocessing Visium Load and preprocessing single cell Find TransferAnchors Transfer annotation

Load Brain data

```
# You need to specify exact directory
setwd("biml_2023")

# We can easily download the data with functions below
options(timeout=600)
InstallData("stxBrain")

brain <- LoadData("stxBrain", type = "anterior1")
Same with
brain = LoadData("stxBrain", type = "anterior1")
```

Explore Seurat object

```
brain
> brain
An object of class Seurat
31053 features across 2696 samples within 1 assay
Active assay: Spatial (31053 features, 0 variable features)
1 image present: anterior1
```

26

Explore metadata

```
brain@meta.data %>% head(3)
```

```
> brain@meta.data %>% head(3)
      orig.ident nCount_Spatial nFeature_Spatial slice region
AAACAAGTATCTCCCA-1 anterior1      13069           4242     1 anterior
AAACACCAATAACTGC-1 anterior1      37448           7860     1 anterior
AAACAGAGCGACTCCT-1 anterior1      28475           6332     1 anterior
```

metadata	explanation
nCount_Spatial	the total number of detected molecules in each sample
nFeature_Spatial	the number of unique genes in each sample
slice	name of the stored image

Explore coordinates

```
brain@images$anterior1@coordinates %>% head(3)
```

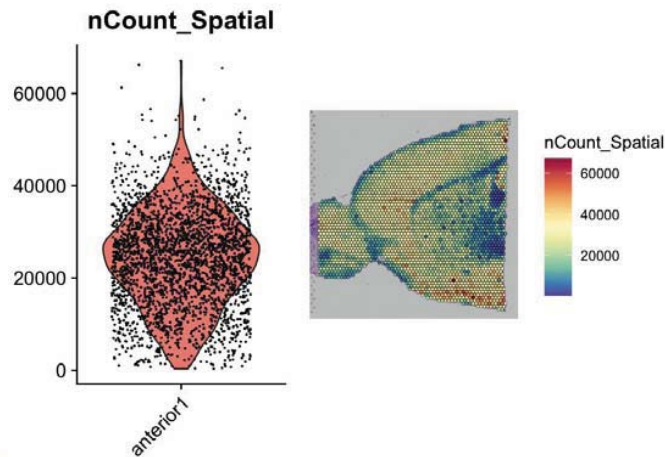
```
> brain@images$anterior1@coordinates %>% head(3)
      tissue row col imagerow imagecol
AAACAAGTATCTCCCA-1  1  50 102    7475    8501
AAACACCAATAACTGC-1  1  59  19    8553    2788
AAACAGAGCGACTCCT-1  1  14  94    3164    7950
```

metadata	explanation
tissue	Binary, indicating if the spot falls inside (1) or outside (0) of tissue
row	The row coordinate of the spot in the array from 0 to 77
col	The column coordinate of the spot in the array
imagerow	The row pixel coordinate of the center of the spot in the full resolution image.
imagecol	The column pixel coordinate of the center of the spot in the full resolution image.

Load Brain data

```
# nCount_Spatial: the total number of detected molecules in each sample
plot1 = VlnPlot(brain, features = "nCount_Spatial", pt.size = 0.1)
+ NoLegend()
plot2 = SpatialFeaturePlot(brain, features = "nCount_Spatial") +
theme(legend.position = "right")

# wrap_plots: take a list of plots and add them into one composition
wrap_plots(plot1, plot2)
```



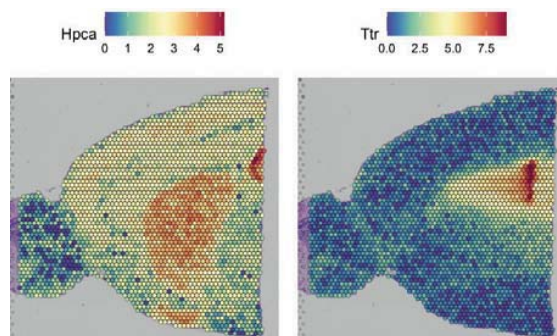
Pre-processing: SCT normalize

```
# SCTransform calculates a model of technical noise using 'regularized negative binomial regression'.
This replaces NormalizeData, ScaleData, and FindVariableFeatures.

brain = SCTransform(brain, assay = "Spatial", verbose = FALSE)
```

Visualization of gene expression

```
SpatialFeaturePlot(brain, features = c("Hpca", "Ttr"))
```



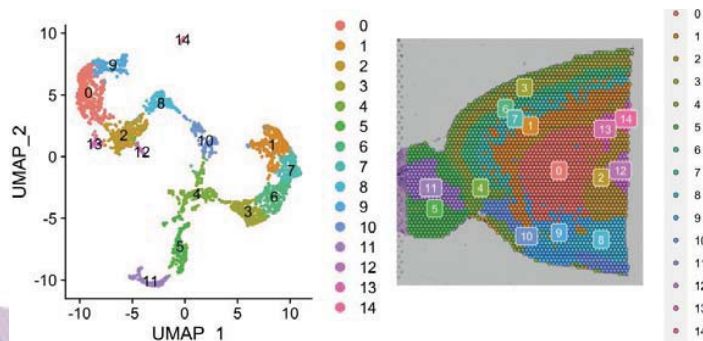
Dimension reduction and clustering

```
# PCA dimensionality reduction
brain = RunPCA(brain, assay = "SCT", verbose = FALSE)

# Compute an SNN (Shared Nearest-neighbor) on the gene expression level
brain = FindNeighbors(brain, reduction = "pca", dims = 1:30)
brain = FindClusters(brain, verbose = FALSE)
brain = RunUMAP(brain, reduction = "pca", dims = 1:30)
```

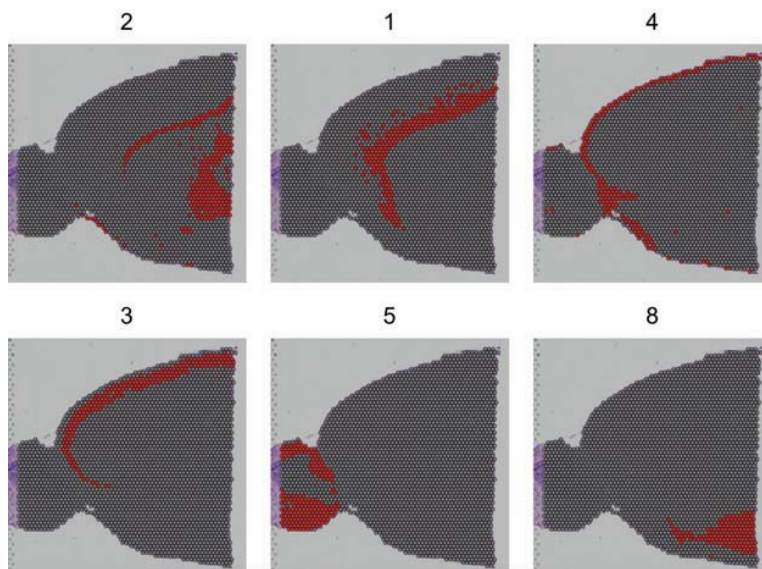
Visualization of each cluster

```
p1 = DimPlot(brain, reduction = "umap", label = TRUE)
p2 = SpatialDimPlot(brain, label = TRUE, label.size = 3)
p1 + p2
```



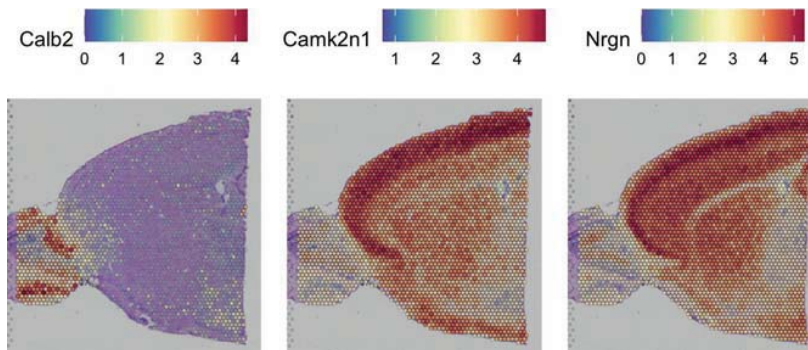
Spatial Dimplot of some cluster

```
# facet.highlight: split each group into its own plot
SpatialDimPlot(brain, cells.highlight = CellsByIdentities(object = brain,
idents = c(2, 1, 4, 3, 5, 8)), facet.highlight = TRUE, ncol = 3)
```



Identification of spatially variable features

```
# We can find cluster 5 markers compared to cluster 6
de_markers = FindMarkers(brain, ident.1 = 5, ident.2 = 6)
SpatialFeaturePlot(object = brain, features =
  rownames(de_markers)[1:3], alpha = c(0.1, 1), ncol = 3)
saveRDS(brain, "./BIML_visium_brain_processed.rds")
```

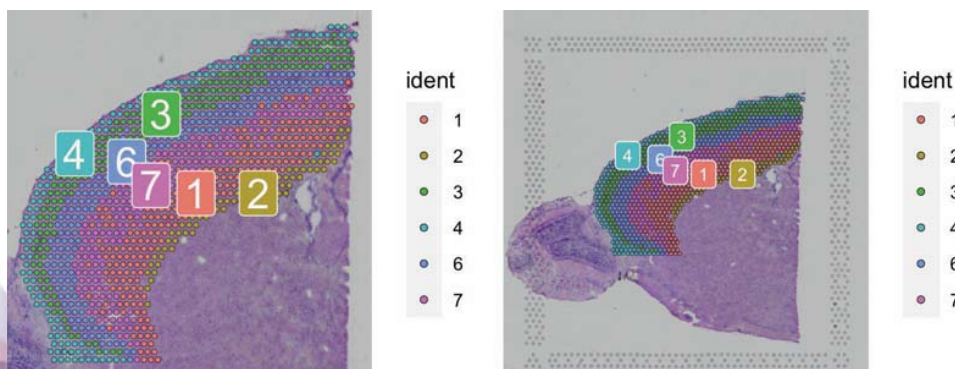


Subset out anatomical regions

```
cortex = subset(brain, idents = c(1, 2, 3, 4, 6, 7))
cortex = subset(cortex, anterior1_imagerow > 400 | anterior1_imagecol < 150, invert = TRUE)
cortex = subset(cortex, anterior1_imagerow > 275 & anterior1_imagecol > 370, invert = TRUE)
cortex = subset(cortex, anterior1_imagerow > 250 & anterior1_imagecol > 440, invert = TRUE)

p1 = SpatialDimPlot(cortex, crop = TRUE, label = TRUE)
p2 = SpatialDimPlot(cortex, crop = FALSE, label = TRUE, pt.size.factor = 1, label.size = 3)
# We merge two plots
p1 + p2

cortex = SCTransform(cortex, assay = "Spatial", verbose = FALSE) %>% RunPCA(verbose = FALSE)
saveRDS(cortex, "./BIML_visium_cortex_processed.rds")
```



Integrate single cell and visium spatial gene expression data

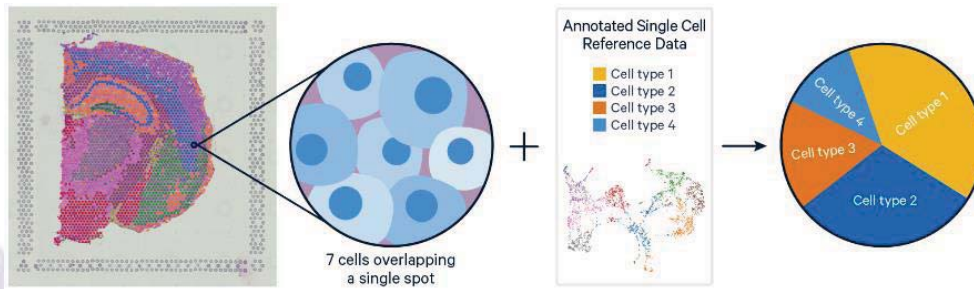
Elucidate spatiality in single cell data and improve resolution in Visium data

Problem

Single Cell RNAseq methods resolve gene expression at the single cell level, but lose the spatial context. Visium spatial gene expression maintains spatial information, but the resolution of each spot is limited (1-10 cells).

Solution

- Deconvolution (Identify the cell types and their relative proportions contributing to a spot)
- Mapping (Assign the most likely dominant cell type to a spot)



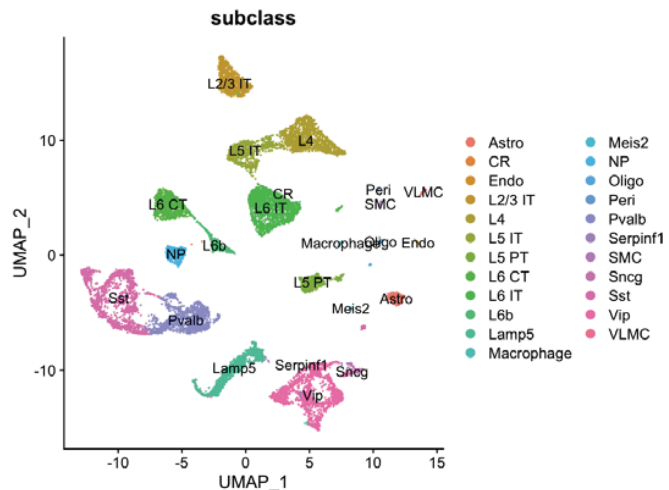
<https://www.10xgenomics.com/resources/analysis-guides/integrating-single-cell-and-visium-spatial-gene-expression-data>

Load reference single cell data

```
allen_reference = readRDS("./BIML_sc_allen_cortex.rds")
allen_reference = SCTransform(allen_reference, ncells = 3000, verbose = FALSE) %>% RunPCA(verbose = FALSE) %>% RunUMAP(dims = 1:30)
allen_reference
```

```
DimPlot(allen_reference, group.by = "subclass", label = TRUE)
```

```
> allen_reference
An object of class Seurat
69225 features across 14249 samples within 2 assays
Active assay: SCT (34608 features, 3000 variable features)
1 other assay present: RNA
2 dimensional reductions calculated: pca, umap
```



Find TransferAnchors

```
# anchors: cross-dataset pairs that are in a matched biological state
# Find a set of anchors between a reference and query object. These anchors can later be used to
  transfer data from the reference to query object using the TransferData object.

anchors = FindTransferAnchors(reference = allen_reference, query =
  cortex, normalization.method = "SCT")
```

Acquire prediction assay

```
# TransferData: Transfer categorical or continuous data across single-cell datasets
# weight.reduction: Dimensional reduction to use for the weighting anchors

predictions.assay = TransferData(anchorset = anchors, refdata =
  allen_reference$subclass, prediction.assay = TRUE, weight.reduction
  = cortex[["pca"]], dims = 1:30)

predictions.assay[,1:3] %>% t()
```

```
> predictions.assay[,1:3] %>% t()
```

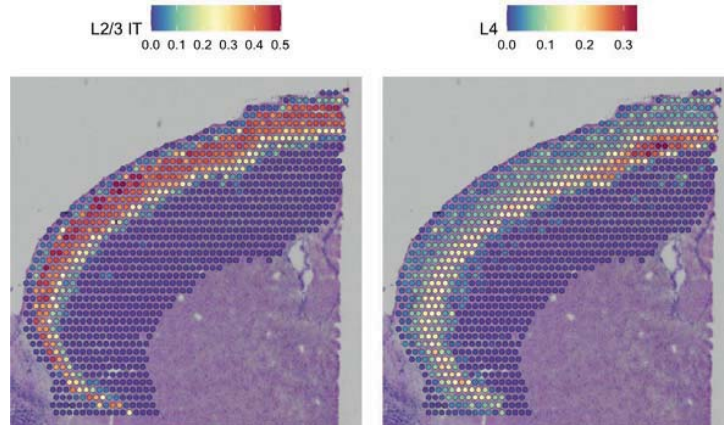
	Vip	Lamp5	Sst	Snca	Serpinf1	Pvalb	Endo	Peri	L6 CT	L6b	L6 IT	L2/3 IT	CR	L5	PT
AAACAGGCGACTCCT-1	0	0	0	0	0	0	0	0	0.00000000	0	0.50591525	0.4422330	0.00000000	0	0
AAACCGGTAGGTACC-1	0	0	0	0	0	0	0	0	0.00000000	0	0.48948994	0.3171848	0.00000000	0	0
AAACGTTCCGTCAGG-1	0	0	0	0	0	0	0	0	0.004239054	0	0.07021274	0.00000000	0.05067488	0	0
	NP	L4	L5 IT	Oligo	Meis2	Astro	Macrophage	VLMC	SMC	max					
AAACAGGCGACTCCT-1	0	0.051851757	0.000000000	0	0	0.0000000	0.0000000	0.0000000	0	0.5059153					
AAACCGGTAGGTACC-1	0	0.113915005	0.079410266	0	0	0.0000000	0.0000000	0.0000000	0	0.4894899					
AAACGTTCCGTCAGG-1	0	0.009649476	0.002329415	0	0	0.299071	0.4430719	0.1207516	0	0.4430719					

Insert the prediction assay into the cortex object

Created prediction assay

```
cortex[["predictions"]] = predictions.assay
DefaultAssay(cortex) = "predictions"
```

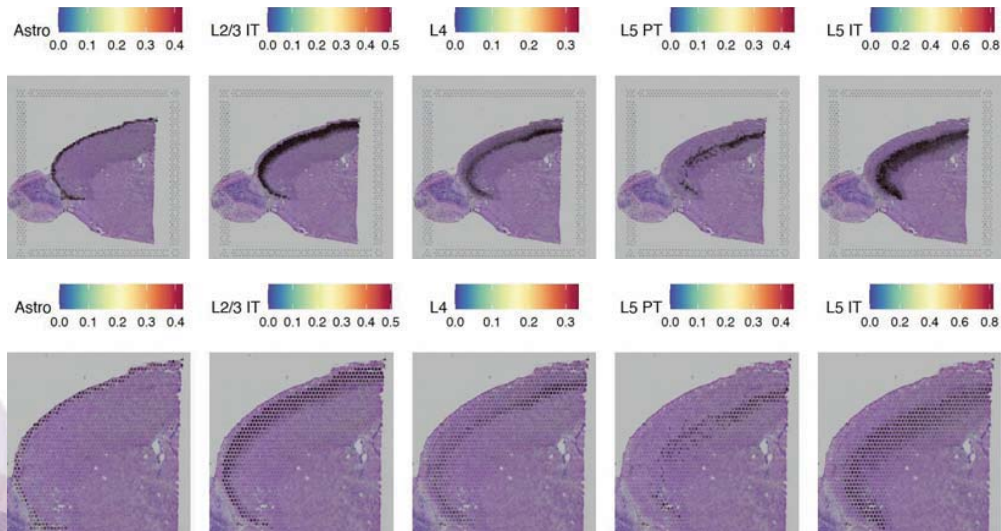
```
SpatialFeaturePlot(cortex, features = c("L2/3 IT", "L4"),
pt.size.factor = 1.6, ncol = 2, crop = TRUE)
```



39

Spatial Feature Plot of predicted cell type proportion

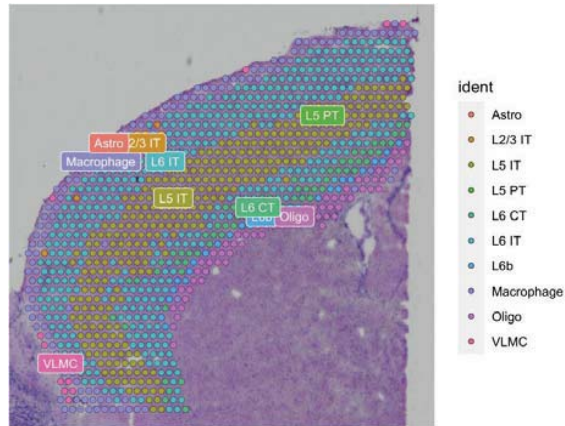
```
pdf('./integrated_spatialfeautrePlot.pdf', width = 10, height = 5)
SpatialFeaturePlot(cortex, features = c("Astro", "L2/3 IT", "L4", "L5 PT", "L5 IT"),
pt.size.factor = 1, ncol = 5, crop = FALSE, alpha = c(0.1, 1))
SpatialFeaturePlot(cortex, features = c("Astro", "L2/3 IT", "L4", "L5 PT", "L5 IT"),
pt.size.factor = 1, ncol = 5, crop = T, alpha = c(0.1, 1))
dev.off()
```



Transfer annotation

We also get predicted cell type metadata for each spot

```
predictions = TransferData(anchorset = anchors, refdata =  
allen_reference$subclass, weight.reduction = cortex[["pca"]], dims = 1:30)  
cortex = AddMetaData(cortex, metadata = predictions)  
cortex$predicted.id <- factor(cortex$predicted.id)  
cortex <- SetIdent(cortex, value="predicted.id")  
SpatialDimPlot(cortex, label = T, label.size = 3)
```



41

Remove all objects before starting next chapter

```
rm(allen_reference)  
rm(brain)  
rm(cortex)  
rm(anchors)  
rm(predictions);  
rm(predictions.assay);  
gc()
```

6. Deconvolution Analysis - SpaceXR

Process single cell for RCTD input Process spatial for RCTD input Run RCTD Process output for deconvolution

RCTD Algorithm

RCTD(**R**obust **c**ell **t**ype **d**ecomposition)

Method that **decomposes cell type mixtures** using cell type profiles learned from single-cell RNA-seq

1. Calculates **the mean gene expression profile of each cell type** within the scRNA-seq reference
2. By fitting each spatial transcriptomics **spot** as a linear combination of individual cell types, RCTD generates a spatial map of cell types
The gene expression of each cell type for a given spot is estimated by fitting a statistical model to observed gene counts, which are assumed to follow Poisson distributions.
3. This model is also optimized with **MLE**

Robust decomposition of cell type mixtures in spatial transcriptomics

Load RCTD input reference dataset (single cell)

```
allen_reference = readRDS("./BIML_sc_allen_cortex.rds")
allen_reference = subset(allen_reference, subclass != c("CR"))
```

Excluded CR cell type since since it needs a minimum of 25 cells for each cell type in the reference

Prepare single cell dataset for RCTD input

```
allen_counts = allen_reference$RNA@counts
allen_meta_data = allen_reference@meta.data
allen_reference = SetIdent(allen_reference, value="subclass")
# Renamed the cluster due to prohibited characters "/"
allen_reference = RenameIdents(allen_reference, "L2/3 IT" = "L2_3 IT")
allen_reference$subclass = allen_reference@active.ident
allen_annotation = allen_reference@meta.data$subclass

names(allen_annotation) = rownames(allen_meta_data)
allen_annotation = as.factor(allen_annotation)
allen_nUMI = allen_meta_data$nCount_RNA
names(allen_nUMI) = rownames(allen_meta_data)
allen_reference = Reference(allen_counts, allen_annotation, allen_nUMI)
```

Process spatial dataset for RCTD input

```
brain = readRDS("./BIML_visium_brain_processed.rds")
spatial_counts_brain = brain@assays$Spatial@counts
spatial_nUMI_brain = colSums(spatial_counts_brain)
coords_brain = brain@images$anterior1@coordinates[,c("col", "row")]
puck_brain = SpatialRNA(coords_brain, spatial_counts_brain, spatial_nUMI_brain)
```

Running RCTD (Full mode, Doublet mode)

```
# Doublet mode: Fits at most two cell types per pixel. Classifies each pixel as 'singlet' or 'doublet' and searches for the cell
types on the pixel.
# Full mode: Fit any number of cell types on each pixel
# Each process takes 30 min ~ 1 hour
```

```
RCTD_brain = create.RCTD(puck_brain, allen_reference, max_cores = 8)
RCTD_brain = run.RCTD(RCTD_brain, doublet_mode = 'full')
RCTD_brain_doublet = run.RCTD(RCTD_brain, doublet_mode = 'doublet')
```

Save RCTD results file

```
# We need to normalize the decomposed matrix
RCTD_results = sweep(RCTD_brain@results$weights, 1,
                    rowSums(RCTD_brain@results$weights), '/')

write.csv(RCTD_results, "./BIML_spacexr_cortex_fullmode.csv")
write.csv(RCTD_brain_doublet@results$results_df, "./BIML_spacexr_cortex_doubletmode.csv")
```

Process RCTD decomposed file (full mode)

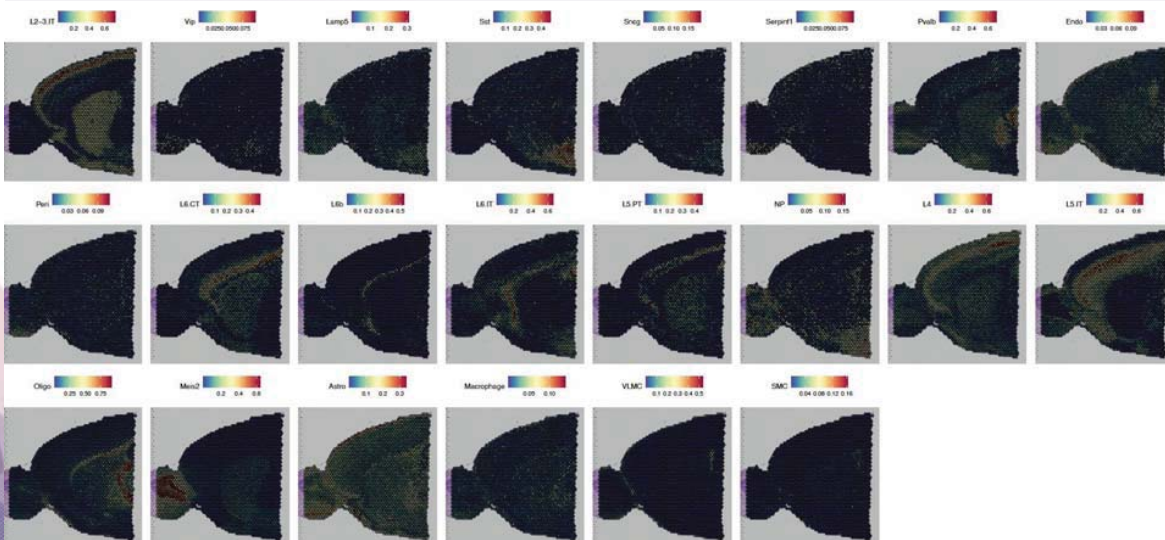
```
full_rctd = read.csv("./BIML_spacexr_cortex_fullmode.csv", header = TRUE,
row.names = 1)
colnames(full_rctd) = gsub("weights.", "", colnames(full_rctd))
brain[['RCTD']] = CreateAssayObject(counts = t(as.matrix(full_rctd)))
DefaultAssay(brain) = "RCTD"
```

Process RCTD decomposed file (doublet mode)

```
doublet_rctd <- read.table("./BIML_spacexr_cortex_doubletmode.csv", sep="," ,
header=TRUE)
spacexr_metadata <- doublet_rctd[,c("X", "first_type")]
colnames(spacexr_metadata) <- c("barcodes", "spacexr_first_type")
rownames(spacexr_metadata) <- spacexr_metadata$barcodes;
spacexr_metadata$barcodes <- NULL
brain <- AddMetaData(brain, spacexr_metadata)
brain <- SetIdent(brain, value="spacexr_first_type")
```

Spatial map of predicted cell type proportions by RCTD

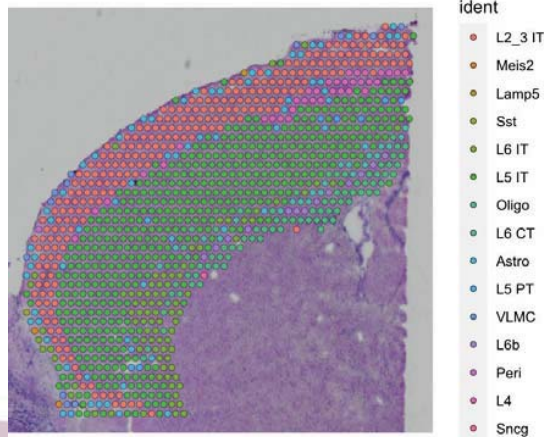
```
pdf("./RCTD_decomposed_Spatial_Featureplot.pdf", height=15, width=25)
SpatialFeaturePlot(brain, features =rownames(brain),ncol=8,
pt.size.factor = 1.6, crop = TRUE)
dev.off()
```



Spatial map of predicted cell type proportions by RCTD

```
cortex = subset(brain, seurat_clusters %in% c(1, 2, 3, 4, 6, 7))
cortex = subset(cortex, anterior1_imagerow > 400 | anterior1_imagecol < 150, invert = TRUE)
cortex = subset(cortex, anterior1_imagerow > 275 & anterior1_imagecol > 370, invert = TRUE)
cortex = subset(cortex, anterior1_imagerow > 250 & anterior1_imagecol > 440, invert = TRUE)
SpatialDimPlot(cortex)
```

```
# This file will be used in the Squidpy and CellChat Analysis
saveRDS(cortex, "./BMIL_visium_cortex_annotated.rds")
```



Remove all objects before starting next chapter

```
rm(brain)
rm(spatial_counts_brain)
rm(spatial_nUMI_brain)
rm(coords_brain)
rm(nUMI_brain)
rm(puck_brain)
rm(barcodes_brain)
rm(RCTD_brain)
rm(RCTD_brain_doublet)
rm(RCTD_results)
rm(full_rctd)
rm(doublet_rctd)
rm(spacexr_metadata)
rm(cortex)
gc()
```


7. Neighborhood Analysis of Co-occurrence - Squidpy

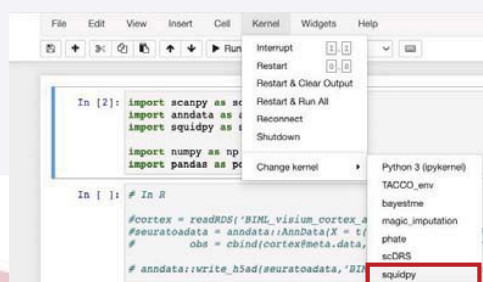
Install squidpy Convert /Load data Neighborhood enrichment analysis Co-occurrence analysis

Install Squidpy via conda

We assume that conda is already installed on your laptop/computer

Instruction below starts from installing jupyter notebook. If you have equivalent platform for visualizing result plots, **just follow black parts**

1. Open your terminal and move to the directory 'biml_2023' where we downloaded all files for the exercise
2. Type `conda env create -f environment.yml` to create conda environment named squidpy
3. Type `python -m pip install --upgrade pip`
4. Type `python -m pip install jupyter`
5. Type `conda activate squidpy`
6. Type `conda install -c anaconda ipykernel`
7. Type `python -m ipykernel install --user --name=squidpy`
8. Type `conda deactivate`
9. Type `jupyter notebook`
10. Change kernel to squidpy like the image



54

Convert data from Seurat object to Anndata

In R

```
cortex = readRDS('BMIL_visium_cortex_annotated.rds')
seuratoadata = anndata::AnnData(X = t(cortex@assays$SCT@data),
                                obs = cbind(cortex@meta.data, cortex@images$anterior1@coordinates),
                                       dtype='float32')

anndata::write_h5ad(seuratoadata, 'BMIL_visium_cortex_anndata.h5ad')
```

55

Load and explore how anndata looks like

In Python

```
import scanpy as sc
import anndata as ad
import squidpy as sq
import numpy as np
import pandas as pd

adata = ad.read_h5ad("BMIL_visium_cortex_anndata.h5ad")
adata
```

AnnData object with $n_obs \times n_vars = 1073 \times 16019$

```
obs: 'orig.ident', 'nCount_Spatial', 'nFeature_Spatial', 'slice', 'region', 'nCount_SCT', 'nFeature_SCT', 'SCT_sn
n_res.0.8', 'seurat_clusters', 'manual_annotation', 'predicted.id', 'prediction.score.Vip', 'prediction.score.Lamp5',
'prediction.score.Sst', 'prediction.score.Sncg', 'prediction.score.Serpinf1', 'prediction.score.Pvalb', 'prediction.s
core.Endo', 'prediction.score.Peri', 'prediction.score.L6.CT', 'prediction.score.L6b', 'prediction.score.L6.IT', 'pre
diction.score.L2.3.IT', 'prediction.score.CR', 'prediction.score.L5.PT', 'prediction.score.NP', 'prediction.score.L
4', 'prediction.score.L5.IT', 'prediction.score.Oligo', 'prediction.score.Meis2', 'prediction.score.Astro', 'predicti
on.score.Macrophage', 'prediction.score.VLMC', 'prediction.score.SMC', 'prediction.score.max', 'spacexr_first_type',
'tissue', 'row', 'col', 'imagerow', 'imagecol'
```

n_obs: cell number
n_vars: gene number
obs: info of seurat@metadata

56

Save spatial information in adata.obsm

In Python

make an array having spatial coordinates

```
spatial_info = adata.obs[['row', 'col']].values.tolist()
adata.obsm['spatial'] = np.array(spatial_info)
```

adata

```
AnnData object with n_obs × n_vars = 1073 × 16019
  obs: 'orig.ident', 'nCount_Spatial', 'nFeature_Spatial', 'slice', 'region', 'nCount_SCT', 'nFeature_SCT', 'SCT_sn
n_res.0.8', 'seurat_clusters', 'manual_annotation', 'predicted.id', 'prediction.score.Vip', 'prediction.score.Lamp5',
'prediction.score.Sst', 'prediction.score.Sncg', 'prediction.score.Serpinf1', 'prediction.score.Pvalb', 'prediction.s
core.Endo', 'prediction.score.Peri', 'prediction.score.L6.CT', 'prediction.score.L6b', 'prediction.score.L6.IT', 'pre
diction.score.L2.3.IT', 'prediction.score.CR', 'prediction.score.L5.PT', 'prediction.score.NP', 'prediction.score.L
4', 'prediction.score.L5.IT', 'prediction.score.Oligo', 'prediction.score.Meis2', 'prediction.score.Astro', 'predicti
on.score.Macrophage', 'prediction.score.VLMC', 'prediction.score.SMC', 'prediction.score.max', 'spacexr_first_type',
'tissue', 'row', 'col', 'imagerow', 'imagecol'
  obsm: 'spatial'
```

57

Builds a spatial graph

In Python

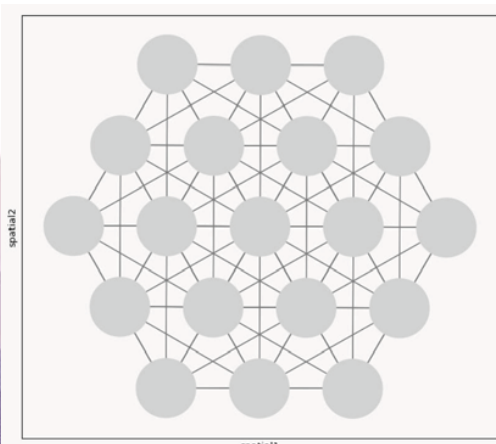
```
sq.gr.spatial_neighbors(adata)
```

Adjacency matrix

```
adata.obsp["spatial_connectivities"]
```

Weighted Adjacency matrix

```
adata.obsp["spatial_distances"]
```



Build spatial graph with observations(= spots) as nodes and neighbor-hood relations between observations as edges.

To identify neighbors, spatial coordinates of spots are used.

58

Calculate enrichment score based on spatial graph

In Python

```
sq.gr.nhood_enrichment(adata, cluster_key="spacexr_first_type")
```

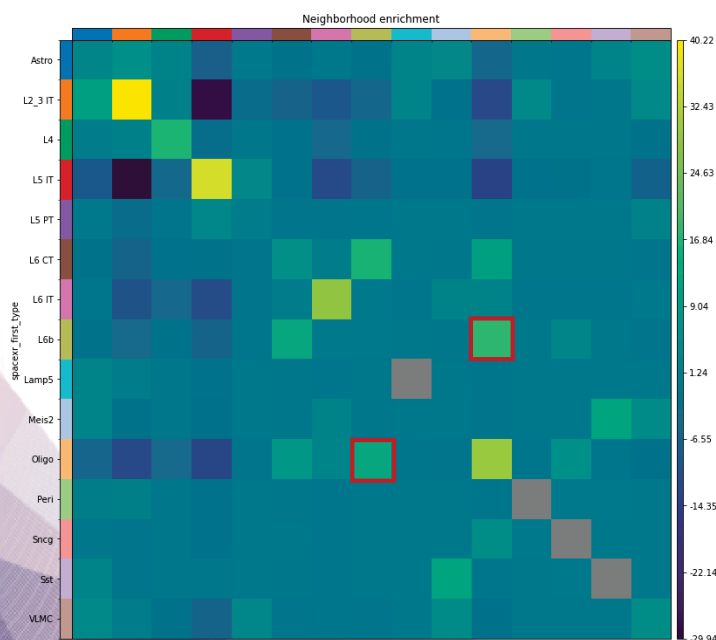
Enrichment score is calculated based on permutation-based test **involving spatial graph**.

If spots belonging to two different clusters are often close to each other, then they will have a high score and can be defined as being *enriched*.

Visualize neighborhood enrichment and select the pair

In Python

```
sq.pl.nhood_enrichment(adata, cluster_key="spacexr_first_type")
```



Except for diagonal squares, Oligo – L6b pair has the high enrichment. (red boxed)

Calculate co-occurrence probability

In Python

```
sq.gr.co_occurrence(adata, cluster_key="spacexr_first_type")
```

Co-occurrence also shows the closeness of two different clusters but **without involving spatial graph**. It is operated on the original spatial coordinates.

The co-occurrence score is defined as:

$$\frac{p(\text{exp}|\text{cond})}{p(\text{exp})}$$

The score is computed across increasing radii size around each spots in the tissue.

$p(\text{exp}|\text{cond})$: conditional probability of observing a cluster exp conditioned on the presence of a cluster cond

$p(\text{exp})$: probability of observing exp in the radius size of interest.

61

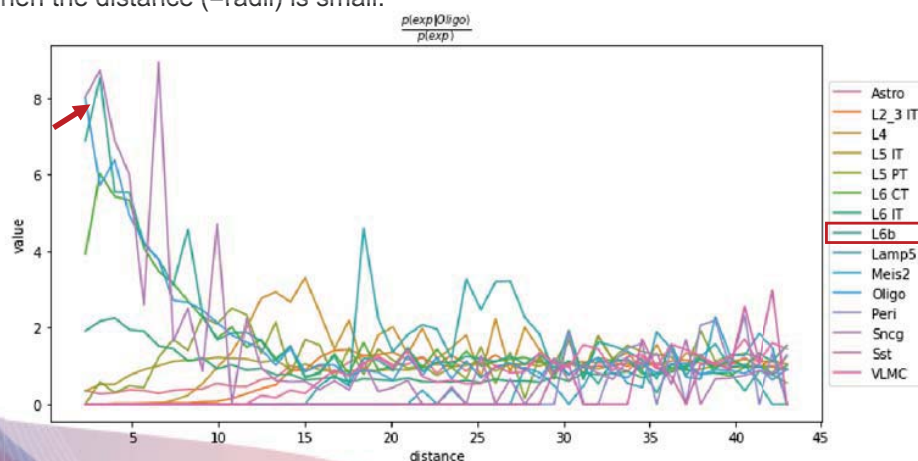
Visualizing Co-occurrence of other cell types with Oligodendrocytes

In Python

```
# Set cond as Oligo by param 'clusters'
sq.pl.co_occurrence(adata, cluster_key="spacexr_first_type",
                    clusters="Oligo", figsize=(8*1.2, 4*1.2))
```

Visualizing the result of Co-occurrence score with Oligodendrocytes.

Similar to neighborhood enrichment result, though not the highest, Oligo shows high Co-occurrence score when the distance (=radii) is small.



62

7. 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.



Load data

```
# Load cell type annotated visium data and set levels for the downstream analysis.
visium.brain = readRDS('./BIML_visium_cortex_annotated.rds')

visium.brain$spacexr_first_type = factor(visium.brain$spacexr_first_type,
levels=c("Astro", "L2_3 IT", "L4", "L5 IT", "L5 PT", "L6 CT", "L6 IT",
"L6b", "Lamp5", "Meis2", "Oligo", "Peri", "Sncg", "Sst", "VLMC"))

Idents(visium.brain) = visium.brain$spacexr_first_type

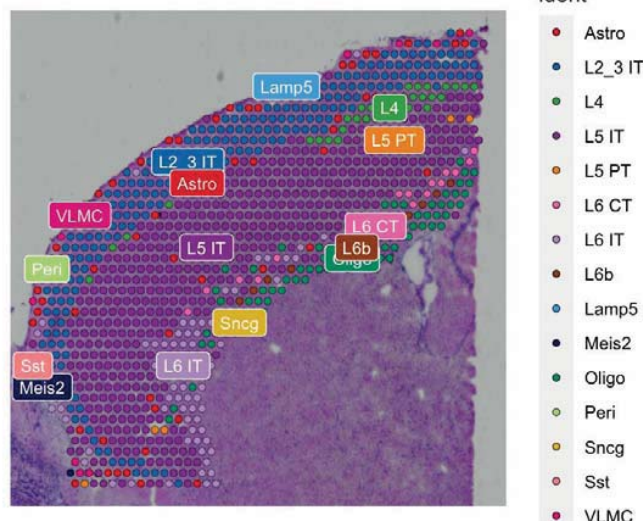
colors = scPalette(nlevels(visium.brain))
names(colors) = c("Astro", "L2_3 IT", "L4", "L5 IT", "L5 PT", "L6 CT", "L6
IT", "L6b", "Lamp5", "Meis2", "Oligo", "Peri", "Sncg", "Sst", "VLMC")
```

65

Visualization of our data

```
SpatialDimPlot(visium.brain, label = T, label.size = 3, cols = colors)
```

There are 15 cell types from our visium data.



66

Prepare input data for CellChat analysis

```
data.input = GetAssayData(visium.brain, slot = "data", assay = "SCT")
meta = data.frame(labels = Idents(visium.brain), row.names =
names(Idents(visium.brain)))
```

```
# check the cell labels
unique(meta$labels)
```

```
> unique(meta$labels) # check the cell labels
[1] L2_3 IT L6 IT Oligo L5 IT L6 CT Astro L5 PT VLNC L6b Peri L4
[12] Meis2 Sncg Lamp5 Sst
15 Levels: Astro L2_3 IT L4 L5 IT L5 PT L6 CT L6 IT L6b Lamp5 Meis2 Oligo Peri Sncg ... VLNC
```

67

Load spatial imaging information

```
# Load spatial imaging information to get the spot information.
```

```
spatial.locs = GetTissueCoordinates(visium.brain, scale = NULL, cols =
c("imagerow", "imagecol"))
```

```
scale.factors = jsonlite::fromJSON(txt = "BIML_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)
```

68

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
```

```
16019 genes.
```

```
1073 cells.
```

```
CellChat analysis of spatial data! The input spatial locations are
```

	x_cent	y_cent
AAACAGAGCGACTCCT-1	3164	7950
AAACCGGGTAGGTACC-1	6517	3407
AAACCGTTCGTCCAGG-1	7715	4371
AAACTCGTGATATAAG-1	4242	9258
AAAGGGATGTAGCAAG-1	4362	5747
AAATAACCATACGGGA-1	3164	7537

69

Set the ligand-receptor interaction database

```
CellChatDB = CellChatDB.mouse
```

```
cellchat@DB = CellChatDB
```

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

Since our toy data is a mouse brain 10x visium data, we load **CellChatDB.mouse**

70

Preprocess of the expression data for cell-cell communication analysis

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

# do parallel
future::plan("multiprocess", workers = 4)

# 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)
```

71

Compute the communication probability and infer cellular communication network

```
# Infers the biologically significant cell-cell communication with permutation test
cellchat = computeCommunProb(cellchat, type = "truncatedMean", trim = 0.1,
                             distance.use = TRUE, interaction.length = 200,
                             scale.distance = 0.01)
cellchat = filterCommunication(cellchat, min.cells = 10)
saveRDS(cellchat, file = 'BIML_visium_cortex_prob_cellchat.rds')

# read RDS file if computeCommunProb() takes too much time (optional)
cellchat = readRDS("BIML_visium_cortex_prob_cellchat.rds")
```

72

Infer the cell-cell communication at a signaling pathway level

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

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

73

Calculate the aggregated cell-cell communication network

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

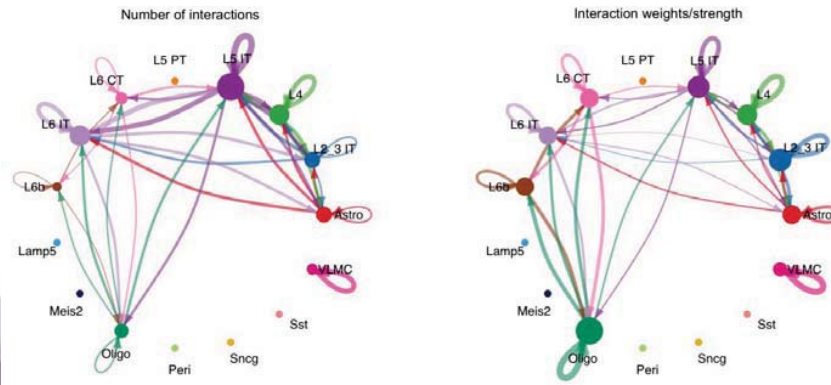
74

Visualization of the aggregated cell-cell communication network

```
par(mfrow = c(1,2), xpd=TRUE)
```

```
netVisual_circle(cellchat@net$count, vertex.weight =
rowSums(cellchat@net$count), weight.scale = T, label.edge= F, title.name =
"Number of interactions")
```

```
netVisual_circle(cellchat@net$weight, vertex.weight =
rowSums(cellchat@net$weight), weight.scale = T, label.edge= F, title.name =
"Interaction weights/strength")
```



Identify ligand-receptor pairs between L6b and Oligo

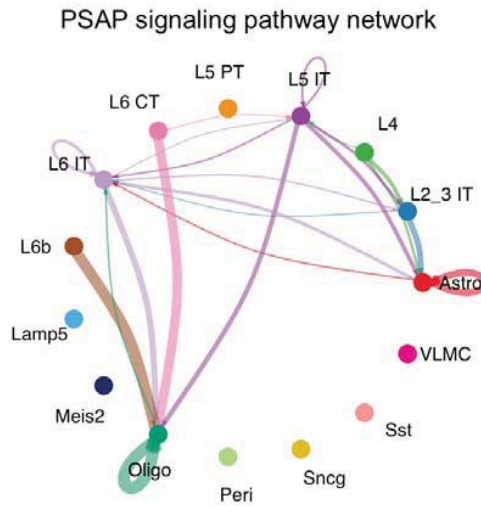
```
CellChat::netVisual_bubble(cellchat, sources.use = c(8), targets.use
=c(11), remove.isolate = FALSE, angle.x = 90, thresh = 0.05) +coord_flip()
```



Levels	Labels	Levels	Labels
1	Astro	9	Lamp5
2	L2_3 IT	10	Meis2
3	L4	11	Oligo
4	L5 IT	12	Peri
5	L5 PT	13	Sncg
6	L6 CT	14	Sst
7	L6 IT	15	VLMC
8	L6b		

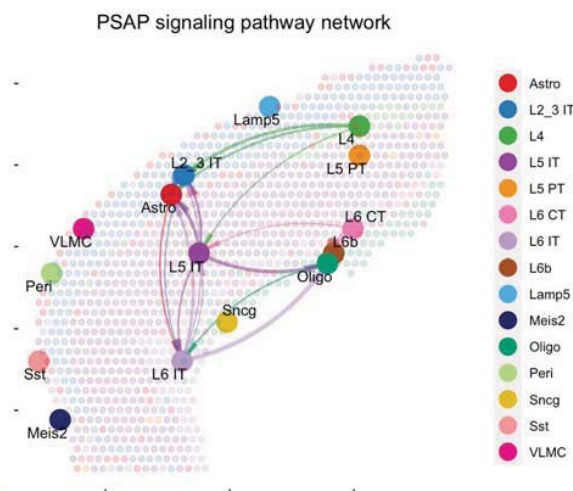
Inferred communication network of signaling pathways – Circle plot

```
pathways.show = c("PSAP")
par(mfrow=c(1,1))
netVisual_aggregate(cellchat, signaling = pathways.show, layout = "circle")
```



Inferred communication network of signaling pathways – Spatial plot

```
par(mfrow=c(1,1))
# Visualization information on the spatial imaging
netVisual_aggregate(cellchat, signaling = pathways.show, layout =
"spatial", edge.width.max = 2, vertex.size.max = 1, alpha.image = 0.2,
vertex.label.cex = 3.5)
```

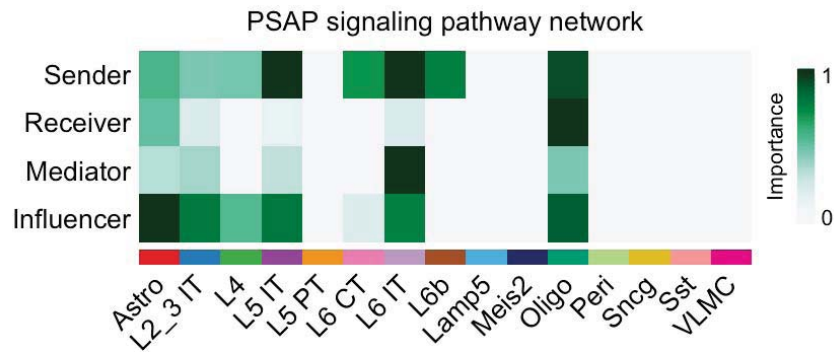


Compute the network centrality scores

```
# Compute the network centrality scores
cellchat = netAnalysis_computeCentrality(cellchat, slot.name = "netP")

# Visualize the centrality score
par(mfrow=c(1,1))
netAnalysis_signalingRole_network(cellchat, signaling = pathways.show,
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

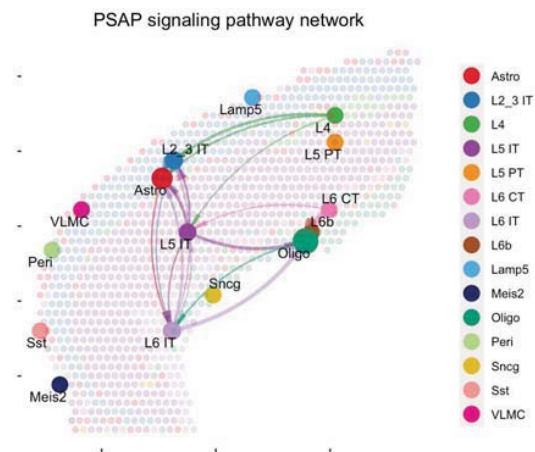


Visualize the network centrality scores - Spatial plot

```
# Visualize the network centrality scores on the spatial imaging.
par(mfrow=c(1,1))
netVisual_aggregate(cellchat, signaling = pathways.show, layout =
"spatial", edge.width.max = 2, alpha.image = 0.2, vertex.weight =
"incoming", vertex.size.max = 3, vertex.label.cex = 3.5)
```

Bigger circle indicates larger incoming signaling.
Incoming and outgoing signal can be visualized.

- Incoming signaling : vertex.weight='incoming'
- Outgoing signaling : vertex.weight='outgoing'



9. Visualization of multi-feature blended spatial plot - STutility

Download spaceranger output Construct STutility object Blended Spatial Feature Plot

Download dataset for STutility

Terminal

```
$ curl -O https://cf.10xgenomics.com/samples/spatial-exp/1.0.0/V1_Mouse_Brain_Sagittal_Anterior/V1_Mouse_Brain_Sagittal_Anterior_filtered_feature_bc_matrix.h5 # download
$ curl -O https://cf.10xgenomics.com/samples/spatial-exp/1.1.0/V1_Mouse_Brain_Sagittal_Anterior/V1_Mouse_Brain_Sagittal_Anterior_spatial.tar.gz
$ tar -xvzf V1_Mouse_Brain_Sagittal_Anterior_spatial.tar.gz
```

Construct STutility object

```
# Set working directory to pre-downloaded directory (optional)
setwd('BIML_visium_STutility_input_files')

infoTable = data.frame(samples =
  c("./V1_Mouse_Brain_Sagittal_Anterior_filtered_feature_bc_matrix.h5"),
  spotfiles = c("./spatial/tissue_positions_list.csv"),
  imgs = c("./spatial/tissue_hires_image.png"),
  json = c("./spatial/scalefactors_json.json"), stringsAsFactors=FALSE)

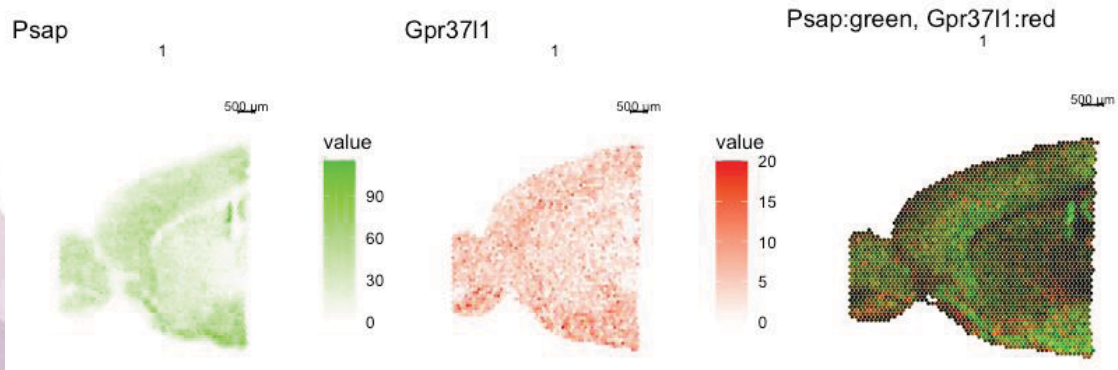
Mouse_Brain_STutility = InputFromTable(infotable = infoTable, platform = "Visium")

# Function used to read HE images in jpeg or png format
Mouse_Brain_STutility = LoadImages(Mouse_Brain_STutility,
  time.resolve = FALSE, verbose = TRUE)
```

Ligand and Receptor Feature Plot

```
plot_grid(
  ST.FeaturePlot(Mouse_Brain_STutility, features = c("Psap"), cols=c("white","green")),
  ST.FeaturePlot(Mouse_Brain_STutility, features = c("Gpr3711"), cols=c("white","red")),
  ST.FeaturePlot(Mouse_Brain_STutility, features = c("Psap","Gpr3711"),
  blend=TRUE,channels.use=c("green","red")), ncol=3)

saveRDS(Mouse_Brain_STutility, "./BIML_visium_cortex_stutility.rds")
```



10. Summary

Summary

10X Genomics Visium sequence data & High-resolution microscope images of tissues

spaceranger count

Gene-Spot matrix

Seurat

Integrative analysis of spatial datasets

Seurat, SpaceXR

Deconvolution and Mapping Analysis

Squidpy

Neighborhood analysis
and co-occurrence

CellChat

Cell-Cell interaction

STutility

Visualization of blended
spatial data

Reference

Space Ranger

Zheng, Grace X.Y., Terry, Jessica M., [...] Bielas, Jason H. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*. 8: 1-12, [doi:10.1038/ncomms14049](https://doi.org/10.1038/ncomms14049)

Seurat

Hao Y, Hao S, Andersen-Nissen E, III WMM, Zheng S, Butler A, Lee MJ, Wilk AJ, Darby C, Zagar M, Hoffman P, Stoeckius M, Papalexi E, Mimitou EP, Jain J, Srivastava A, Stuart T, Fleming LB, Yeung B, Rogers AJ, McElrath JM, Blish CA, Gottardo R, Smibert P, Satija R (2021). "Integrated analysis of multimodal single-cell data." *Cell*. doi:10.1016/j.cell.2021.04.048, <https://doi.org/10.1016/j.cell.2021.04.048>

SpaceXR

Cable, Dylan M., et al. "Robust decomposition of cell type mixtures in spatial transcriptomics." *Nature Biotechnology* 40.4 (2022): 517-526.

Squidpy

Palla, G., Spitzer, H., Klein, M. et al. Squidpy: a scalable framework for spatial omics analysis. *Nat Methods* 19, 171–178 (2022). <https://doi.org/10.1038/s41592-021-01358-2>

CellChat

Jin, S., Guerrero-Juarez, C.F., Zhang, L. et al. Inference and analysis of cell-cell communication using CellChat. *Nat Commun* 12, 1088 (2021). <https://doi.org/10.1038/s41467-021-21246-9>

Stutlity

Bergenström, J., Larsson, L. & Lundeberg, J. Seamless integration of image and molecular analysis for spatial transcriptomics workflows. *BMC Genomics* 21, 482 (2020). <https://doi.org/10.1186/s12864-020-06832-3>

11. Q&A