Acknowledgement of Country

The University of Queensland (UQ) acknowledges the Traditional Owners and their custodianship of the lands on which we meet.

We pay our respects to their Ancestors and their descendants, who continue cultural and spiritual connections to Country.

We recognise their valuable contributions to Australian and global society.

Image: Digital reproduction of *A guidance through time* by Casey Coolwell and Kyra Mancktelow

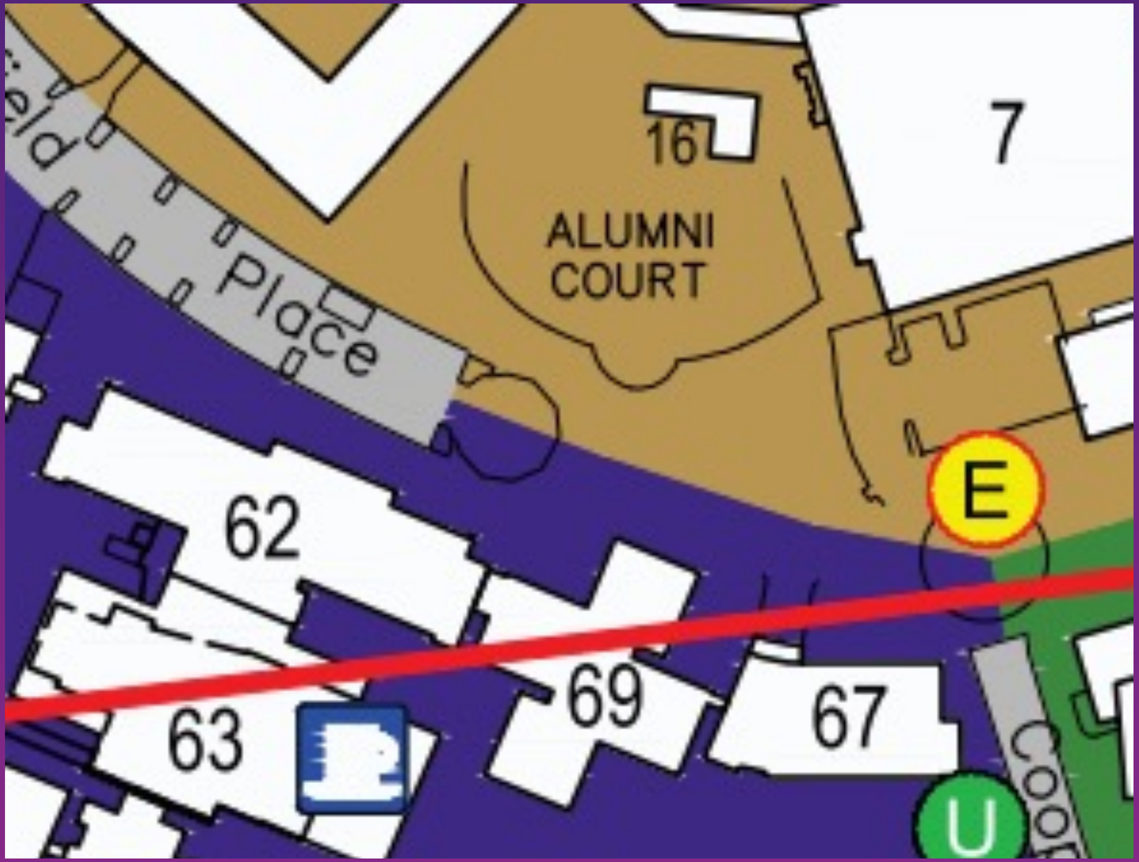
General Information:

- We are currently located in Building 69

E Emergency evacuation point

- Food court and bathrooms are located in Building 63

- If you are experiencing cold/flu symptoms or have had COVID in the last 7 days please ensure you are wearing a mask for the duration of the module



Data Agreement

To maximize your learning experience, we will be working with genuine human genetic data, during this module.

Access to this data requires agreement to the following in to comply with human genetic data ethics regulations

Please email pctgadmin@imb.uq.edu.au with your name and the below statement to confirm that you agree with the following:

“I agree that access to data is provided for educational purposes only and that I will not make any copy of the data outside the provided computing accounts.”

Desktop Access

For non-UQ attendees, you are provided with a registration instruction for a guest account (A4 paper).

After you have completed the online registration, use the provided Username and the Password that you set to log into the desktop.

Cluster Access

- You have all been provided with login details to computing resources needed for the practical component
- An SSH terminal is needed to connect to the computing:
  - Windows: Install PuTTY
    - Hostname: as provided (203.101.228.xxx)
    - User: as provided
    - Check Connection > SSH > X11 > Enable X11 forwarding
  - Mac/Linux: Use the terminal
    - `ssh -X <user>@203.101.228.xxx`
- If interactive R plotting does not work on your machine, you can generate plot on the server and then download
  - Windows: use WinSCP -> enter login information
  - Or use Command Prompt -> sftp <user>@203.101.228.xxx
    - get xxx.pdf and the file will be in your user directory

Module 5 Cellular Transcriptomics

Room 304, Building 69

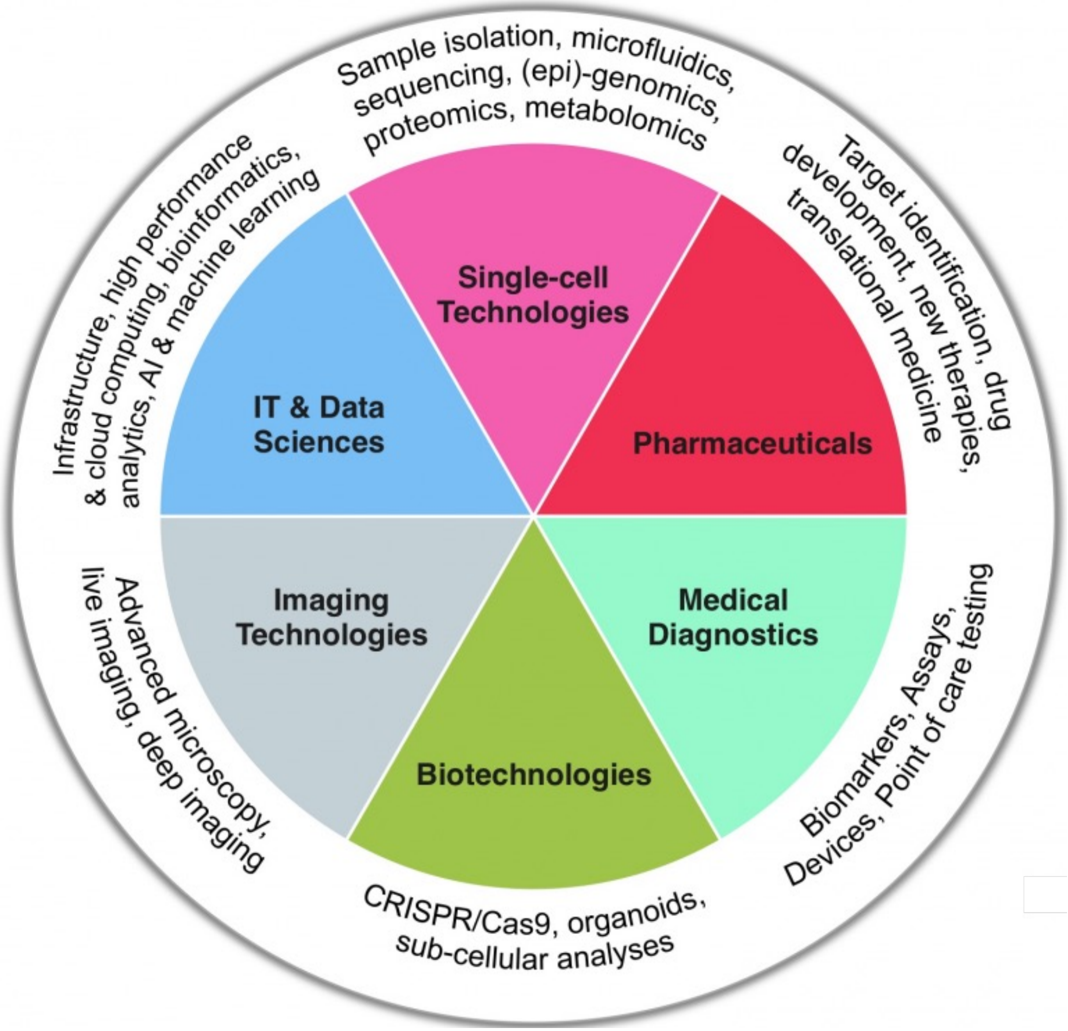
Slides and Practical notes:

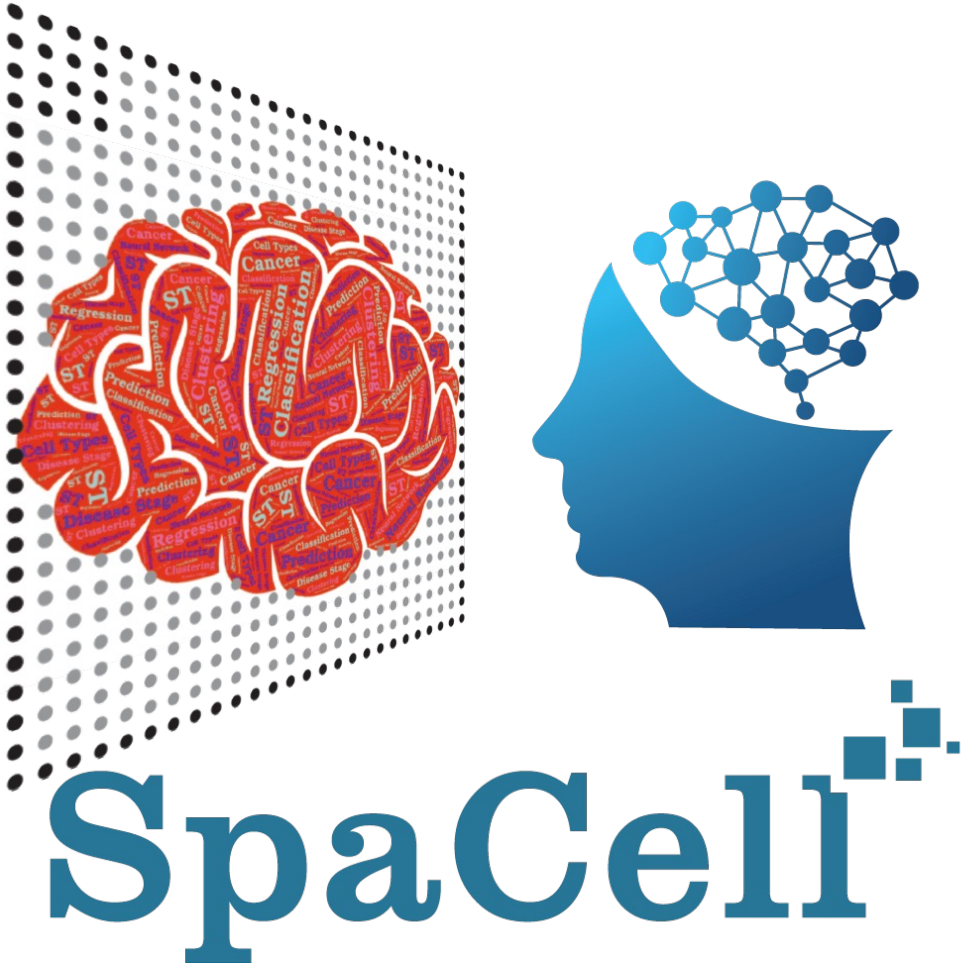
https://cnsgenomics.com/data/teaching/GNGWS22/....TBA.../

Day 2 (June 24th Friday): Spatial transcriptomics analysis

Lecture (Morning; Spatial transcriptomics and machine learning – key concepts)		
9:00-9:15am	Introduction to spatial technologies and applications	Quan Nguyen
9:15-9:30am	Data structure	Duy Pham
9:30-9:45am	Introduction to machine learning: machine learning vs statistical learning vs artificial intelligence in genomics and biological imaging	Quan Nguyen
9:45-10:00am	Introduction to machine learning: key concepts	Quan Nguyen
10:00-10:40am	Machine learning in single cell data	Guiyan Ni
10:40-11:00pm	Break	
11-11:10pm	Spatial transcriptomics analysis – integrating imaging, spatial and gene expression data	Quan Nguyen
11:10-11:30pm	Predicting gene expression using spatial imaging data	Xiao Tan & Quan Nguyen
11:30-11:50pm	Analysis methods to study cell-cell interactions	Duy Pham & Quan Nguyen

Spatial transcriptomics and Machine learning


Sample isolation, microfluidics, sequencing, (epi)-genomics, proteomics, metabolomics
Single-cell Technologies
Target identification, drug development, new therapies, translational medicine
Pharmaceuticals
Biomarkers, Assays, Devices, Point of care testing
Medical Diagnostics
CRISPR/Cas9, organoids, sub-cellular analyses
Biotechnologies
Advanced microscopy, live imaging, deep imaging
Imaging Technologies
Infrastructure, high performance & cloud computing, bioinformatics, analytics, AI & machine learning
IT & Data Sciences

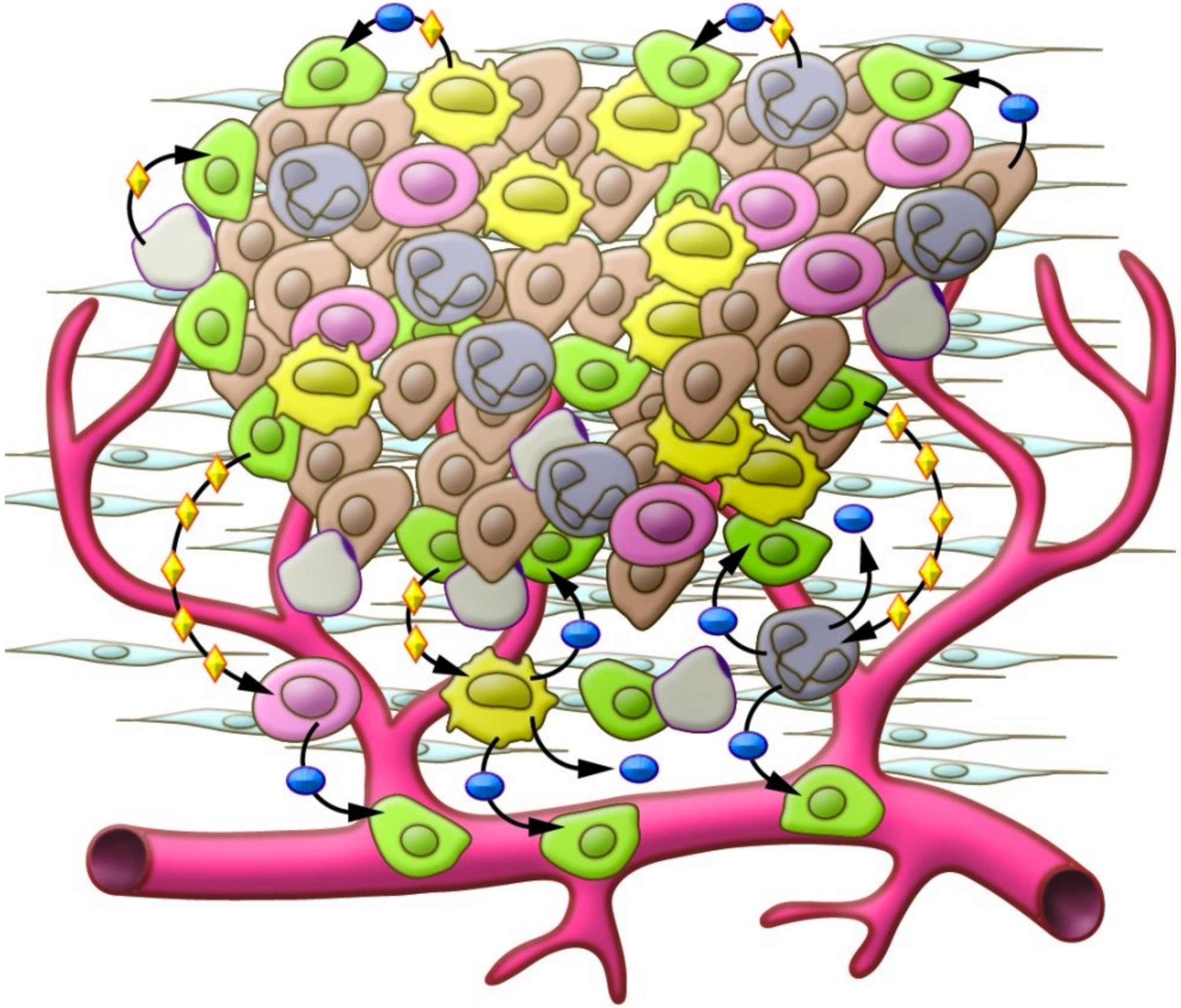


The G&G Cellomics Team

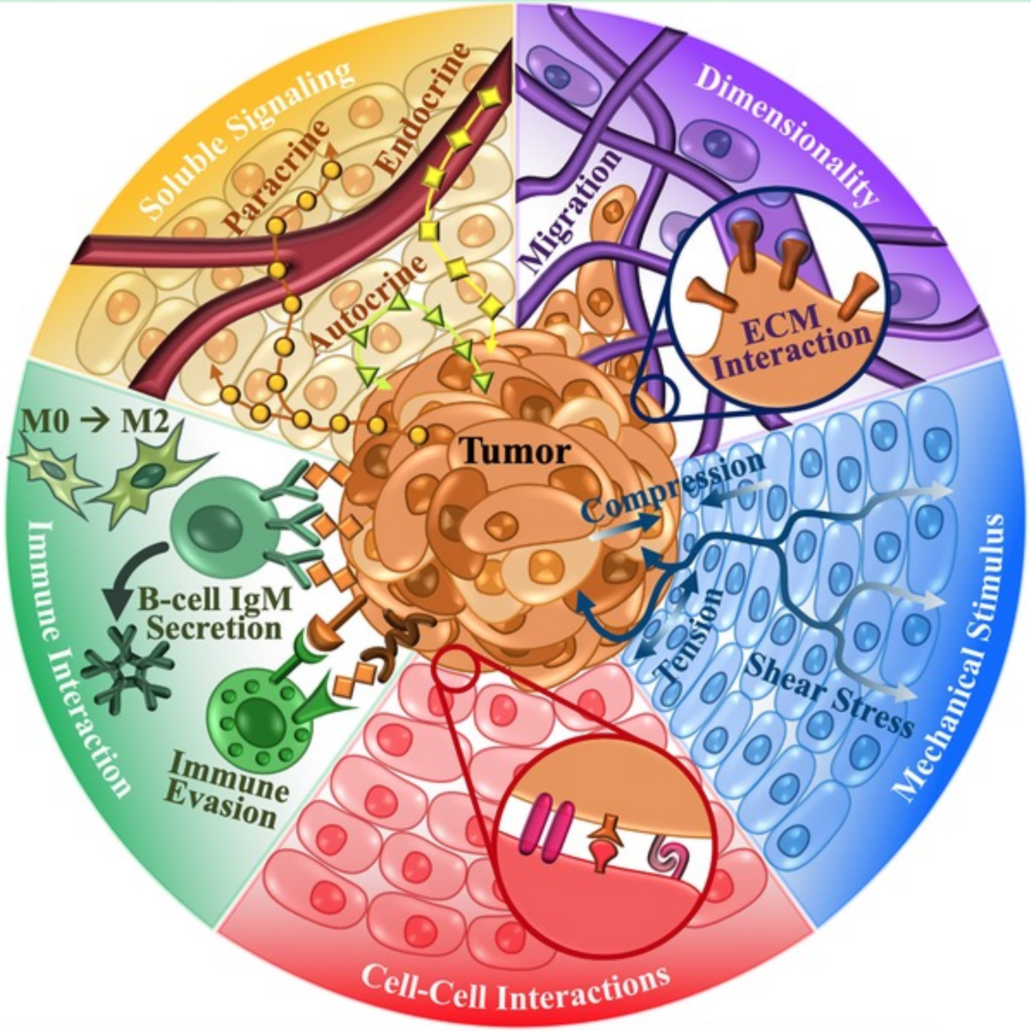
Quan Nguyen, Guiyan Ni, Sally Mortlock, Duy Pham, Xiao Tan

Introduction spatial transcriptomics

Cancer in a native tissue



(Korkaya et al, 2011)



(Bregenzer et al, 2019)

- Cell-type composition and organisation and cell-cell interactions are important
- Complex in vivo processes have direct effects on or are the consequences of transcriptional regulation

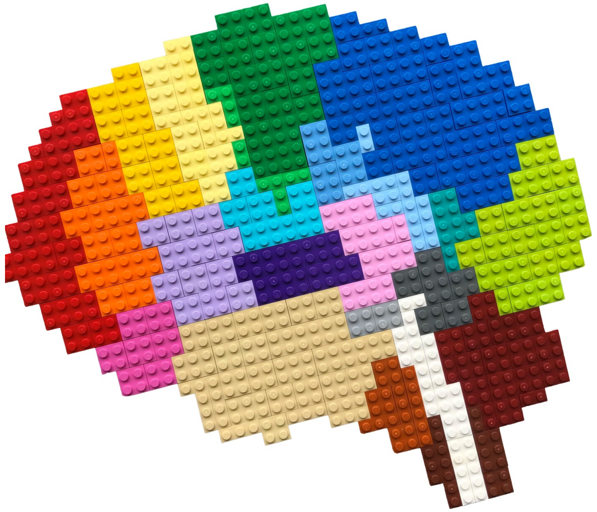
Spatial transcriptomics approach

Bulk

Single cell

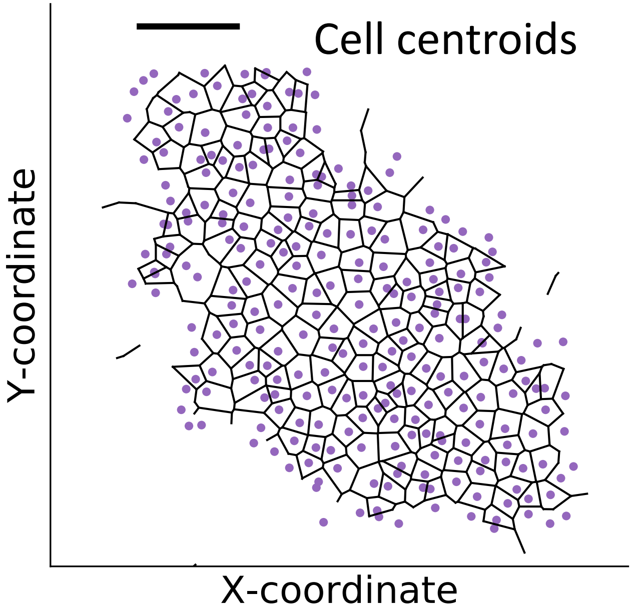
Spatial

Lego:
(@boxia)

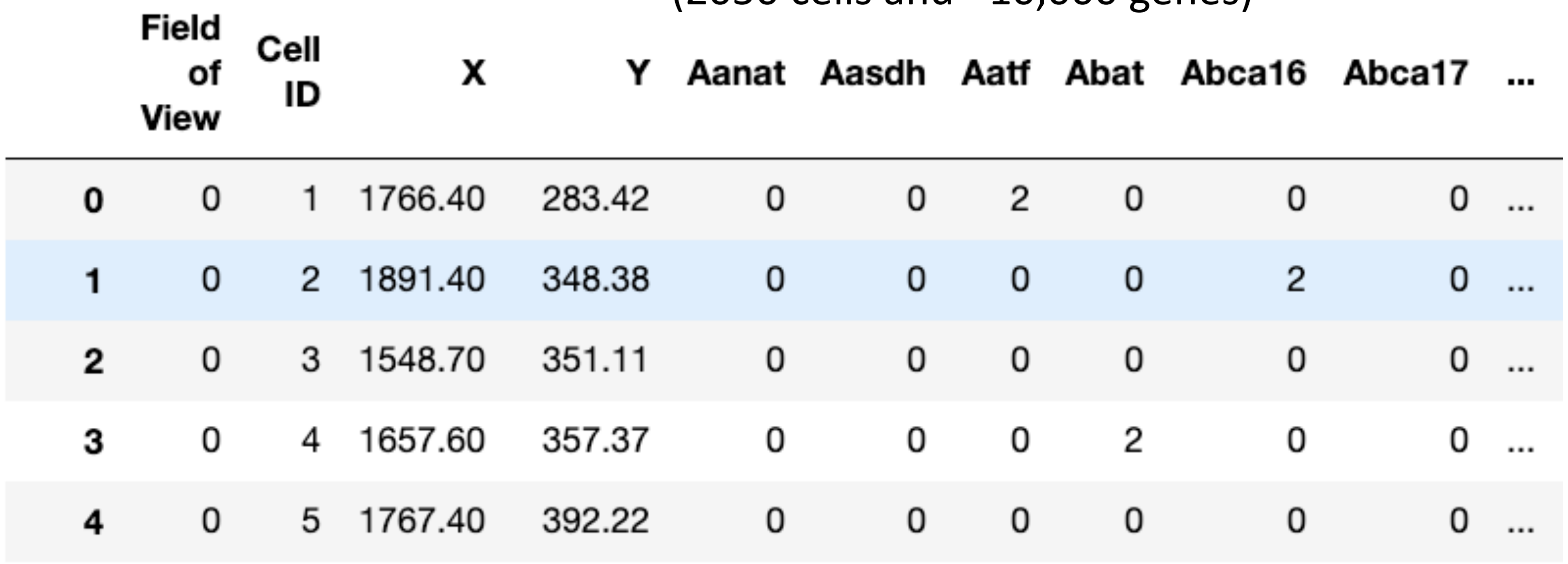


Fruit salad:
(@LGMartelotto)

Spatial Transcriptomics Data (seqFISH): expression + location

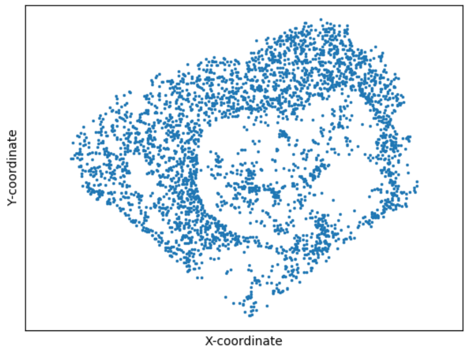


(2050 cells and ~10,000 genes)

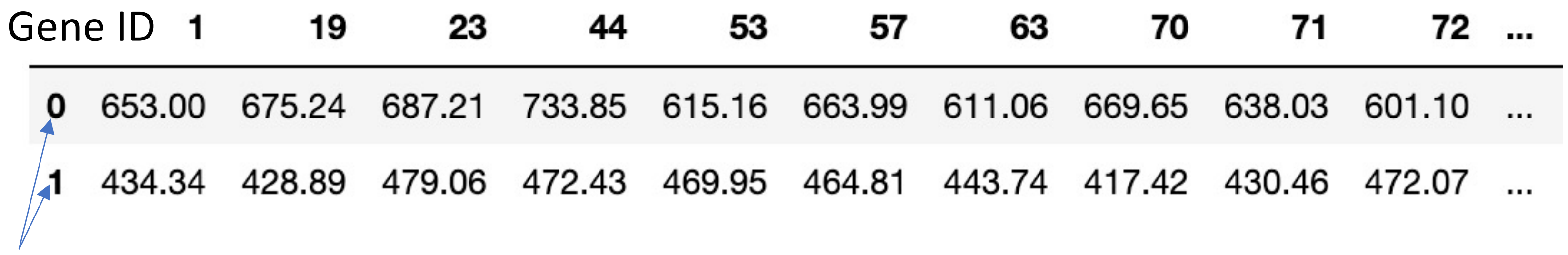


	Field of View	Cell ID	X	Y	Aanat	Aasdh	Aatf	Abat	Abca16	Abca17	...
0	0	1	1766.40	283.42	0	0	2	0	0	0	...
1	0	2	1891.40	348.38	0	0	0	0	2	0	...
2	0	3	1548.70	351.11	0	0	0	0	0	0	...
3	0	4	1657.60	357.37	0	0	0	2	0	0	...
4	0	5	1767.40	392.22	0	0	0	0	0	0	...

Fluorescence single molecule counts



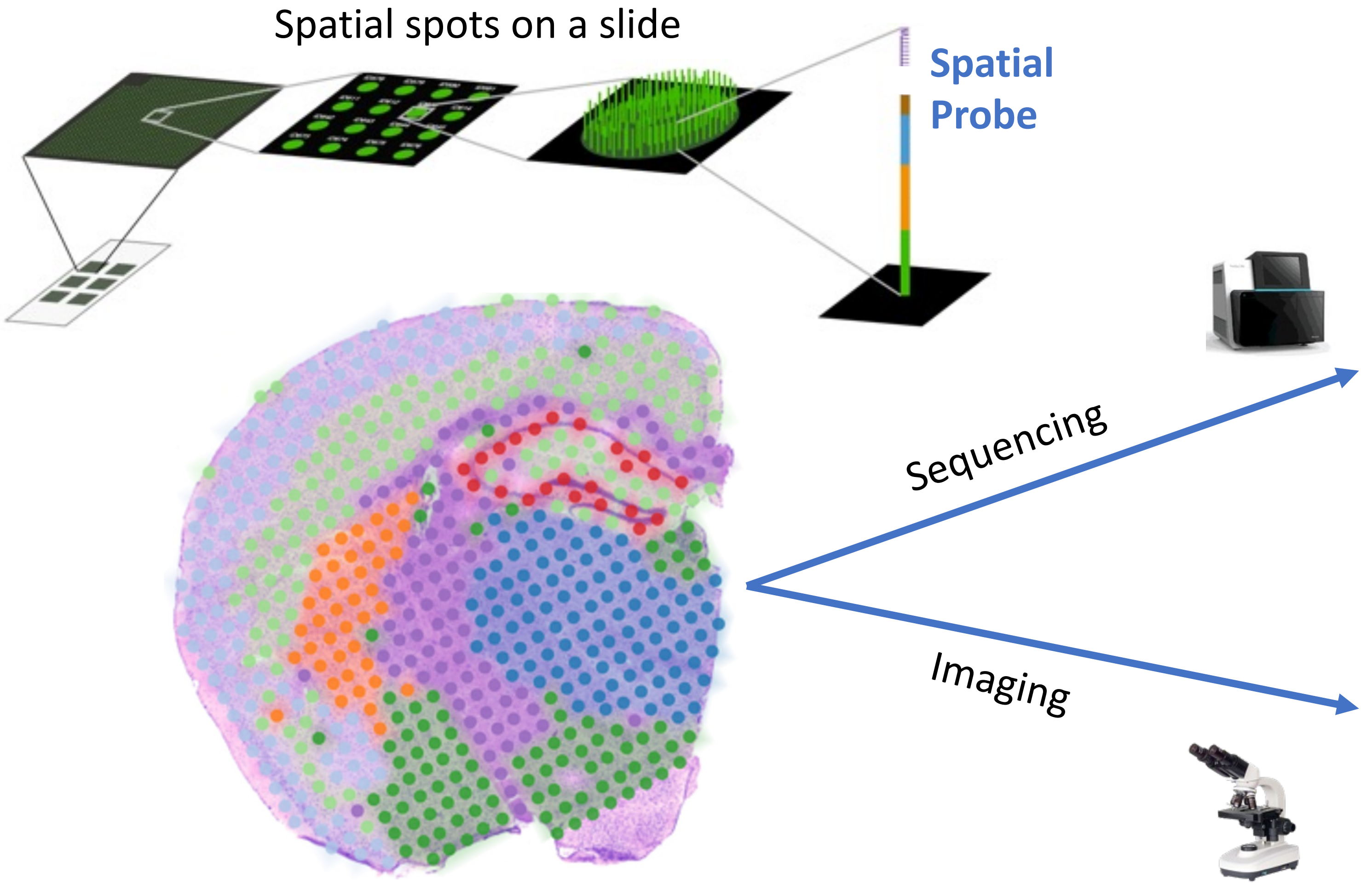
Example of seqFISH RNA in a cell: 3247 genes



Gene ID	1	19	23	44	53	57	63	70	71	72	...
0	653.00	675.24	687.21	733.85	615.16	663.99	611.06	669.65	638.03	601.10	...
1	434.34	428.89	479.06	472.43	469.95	464.81	443.74	417.42	430.46	472.07	...

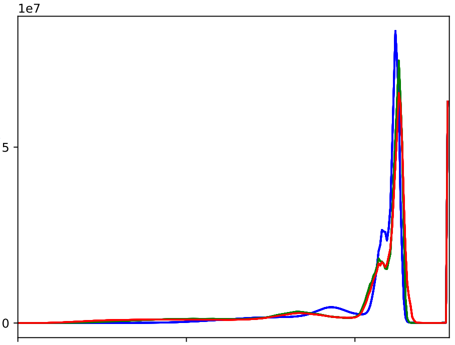
Coordinates

Spatial transcriptomics captures tissue morphology and transcriptome



Spatial Expression

	4x26	3x26	…
Fam234	0	1	…
Nefl	3	0	…
Sema5a	0	1	…
…	…	…	…



Color image intensity

- On-tissue expression profiling (>20,000 genes); each spot contains ~1-9 cells; tissue < 6.5 mm x 6.5 mm
- Other spatial technologies are different (complementary) in resolution, throughput, scale, sensitivity ect.

Data structure of
scRNAseq and Spatial transcriptomics

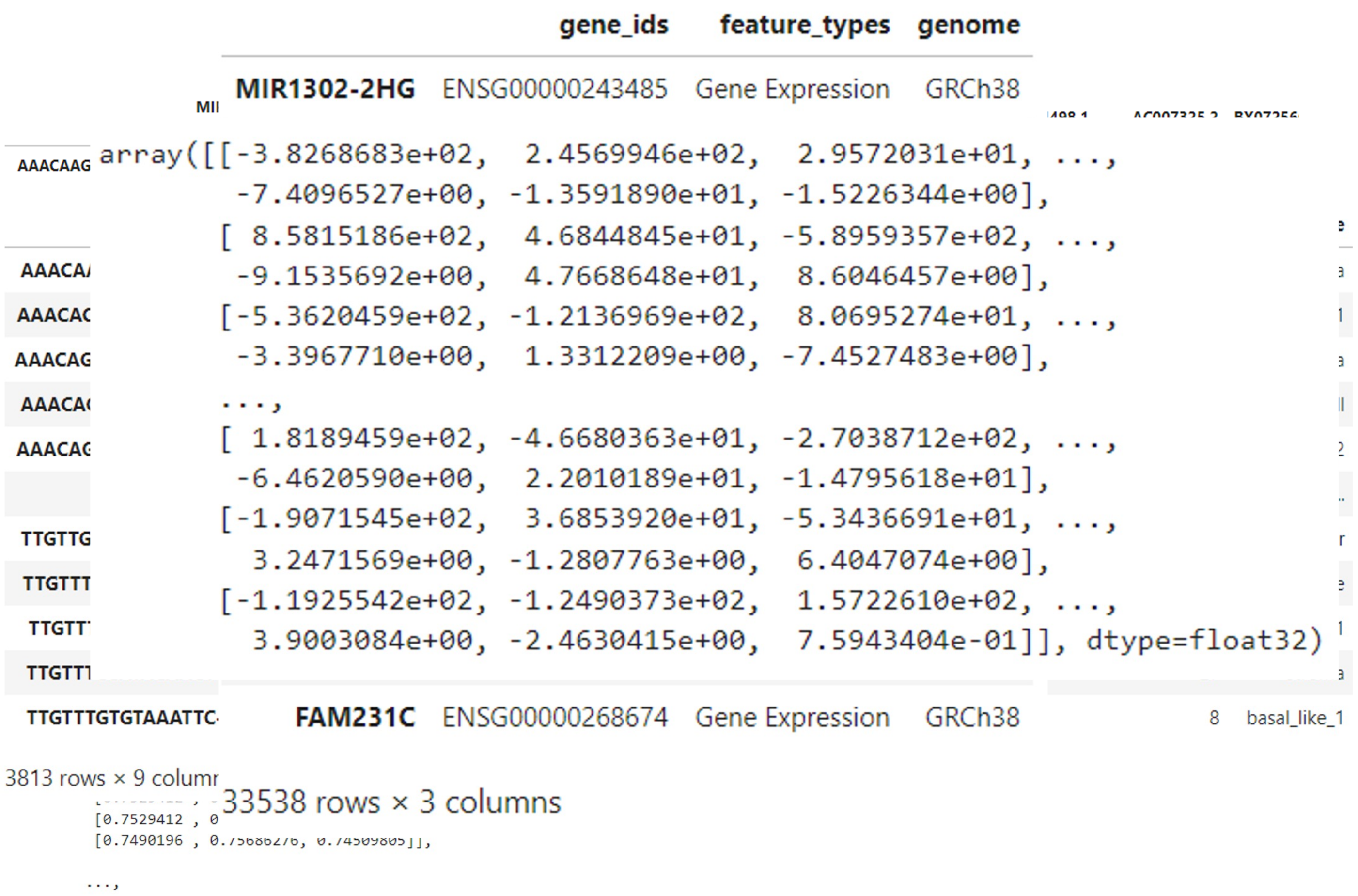
Definition

- **Data:** Collection of raw facts

- **Data structure:** specialized format for ***organizing*** and ***storing*** data in memory that contains not only the ***elements*** stored but also ***their relationship*** to each other

scRNAseq or spatial transcriptomics data

- **Gene expression matrix:**
  - Row: cells/spots
  - Column: genes
- **Cells/spots metadata:**
  - Cell type
  - Batch
  - Spatial coordinates
  - …
- **Genes metadata:**
  - Reference
  - Ensembl ID
  - …
- **Image:**
  - H&E image
- **Embedding**
  - PCA
  - UMAP



	gene_ids	feature_types	genome
MIR1302-2HG	ENSG00000243485	Gene Expression	GRCh38
FAM231C	ENSG00000268674	Gene Expression	GRCh38

33538 rows × 3 columns

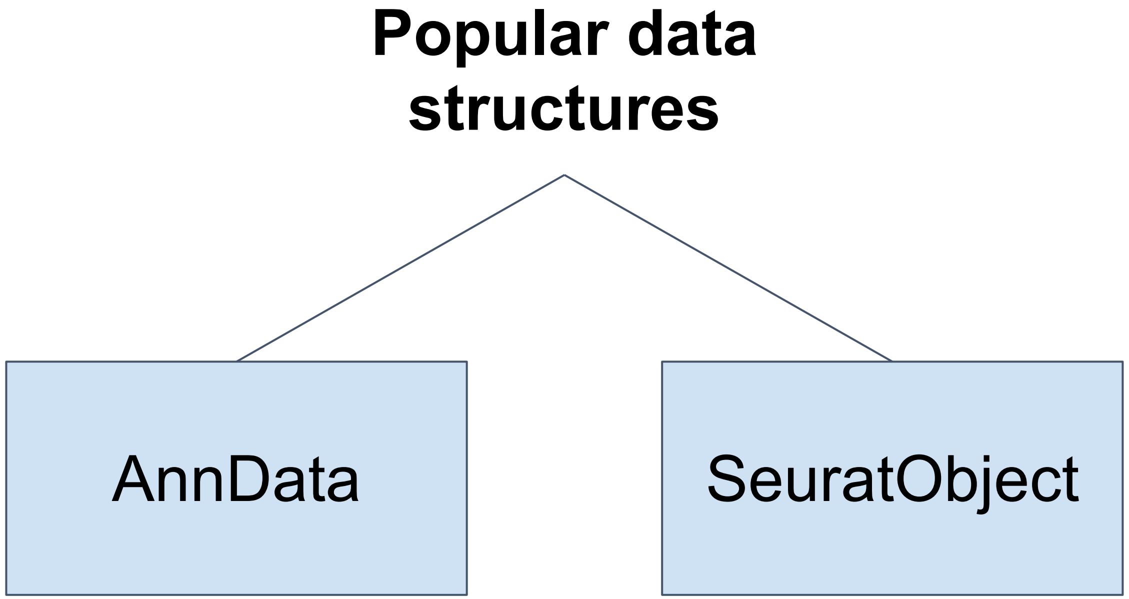
```
array([[-3.8268683e+02,  2.4569946e+02,  2.9572031e+01, ...,
        -7.4096527e+00, -1.3591890e+01, -1.5226344e+00],
       [ 8.5815186e+02,  4.6844845e+01, -5.8959357e+02, ...,
        -9.1535692e+00,  4.7668648e+01,  8.6046457e+00],
       [-5.3620459e+02, -1.2136969e+02,  8.0695274e+01, ...,
        -3.3967710e+00,  1.3312209e+00, -7.4527483e+00],
       ...,
       [ 1.8189459e+02, -4.6680363e+01, -2.7038712e+02, ...,
        -6.4620590e+00,  2.2010189e+01, -1.4795618e+01],
       [-1.9071545e+02,  3.6853920e+01, -5.3436691e+01, ...,
         3.2471569e+00, -1.2807763e+00,  6.4047074e+00],
       [-1.1925542e+02, -1.2490373e+02,  1.5722610e+02, ...,
         3.9003084e+00, -2.4630415e+00,  7.5943404e-01]], dtype=float32)
```

AAACAAG
AAACA
AAACAC
AAACAG
AAACA
AAACAG
TTGTTG
TTGTTT
TTGTT
TTGTT
TTGTTTGTGTAAATTC 8 basal_like_1

3813 rows × 9 columr

```
[0.7529412 , 0
[0.7490196 , 0.75686276, 0.74509805]],
...,
```

Popular data structures

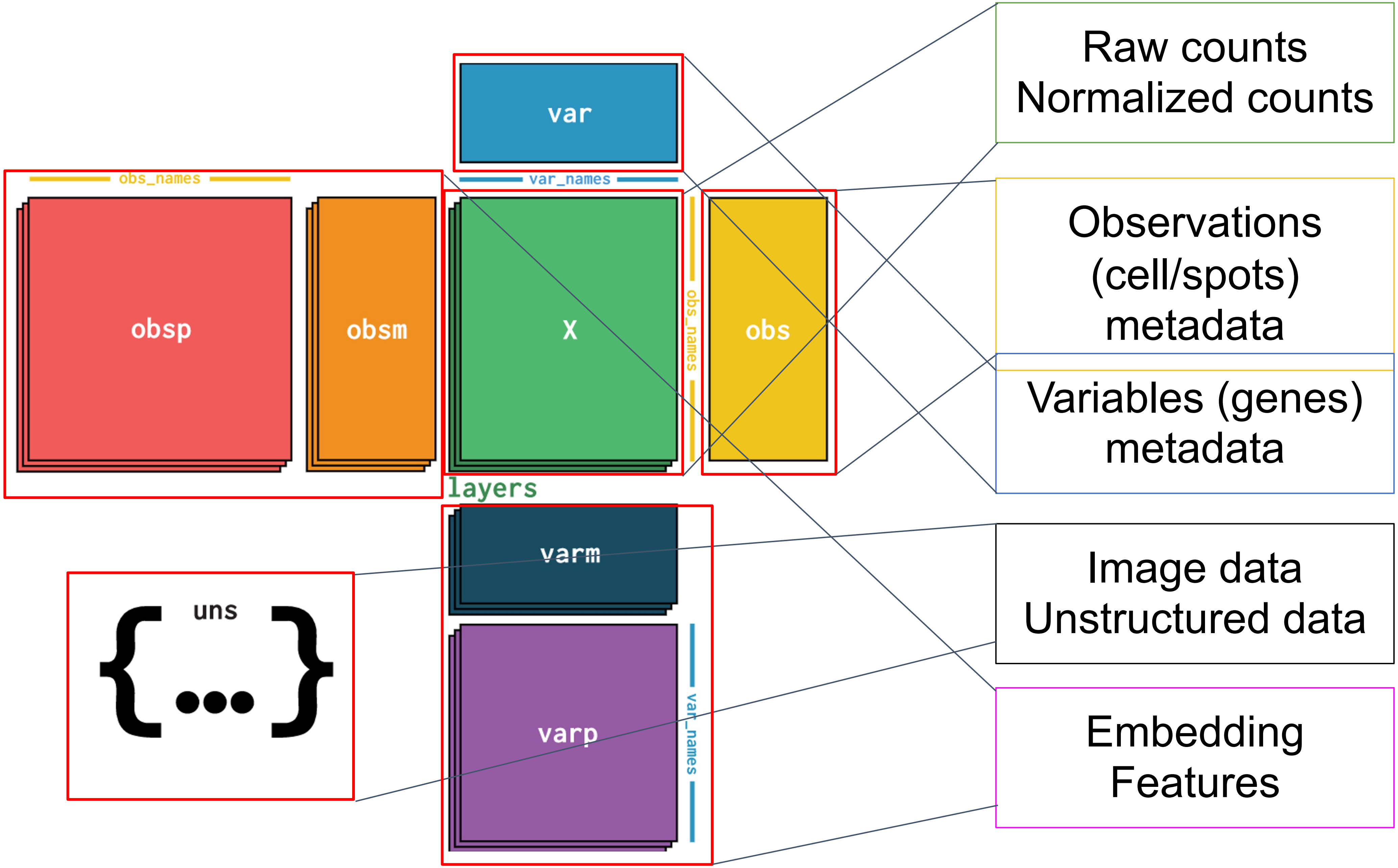


Popular data structures

AnnData

SeuratObject

AnnData (Annotated data) - Python



Raw counts
Normalized counts

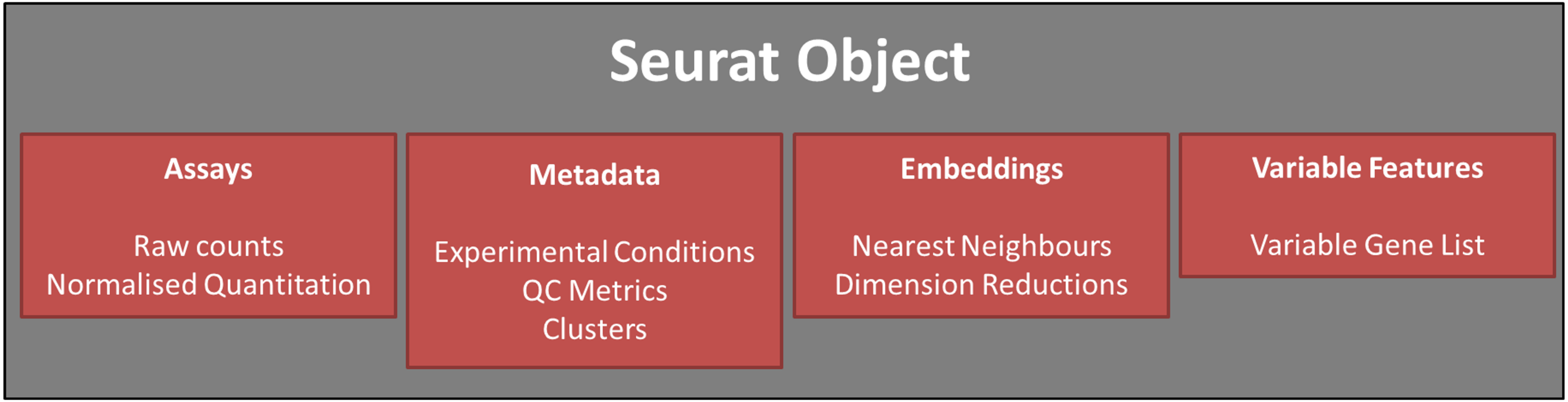
Observations (cell/spots) metadata

Variables (genes) metadata

Image data
Unstructured data

Embedding
Features

SeuratObject - R



Seurat Object

Assays

Raw counts
Normalised Quantitation

Metadata

Experimental Conditions
QC Metrics
Clusters

Embeddings

Nearest Neighbours
Dimension Reductions

Variable Features

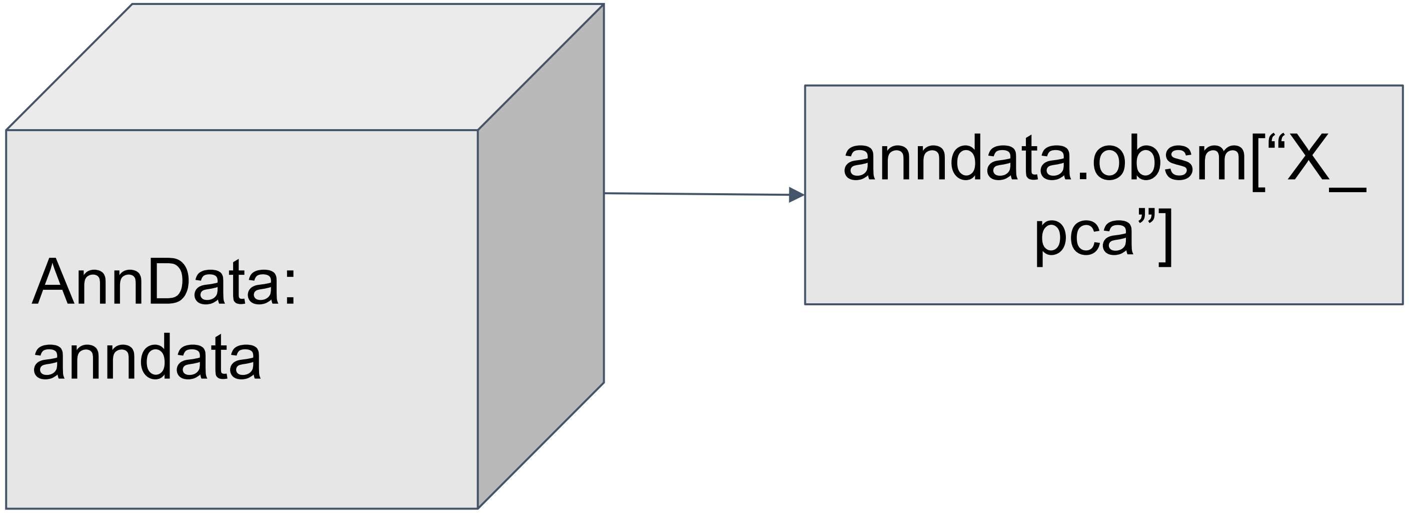
Variable Gene List

Use case:
Perform K-means clustering and store to AnnData

How?

1. Extract the PCs components from AnnData for every cells/spots
2. Using external scikit-learn package for K-means clustering
3. Get the K-means clustering results
4. Add results to observation annotation of AnnData object

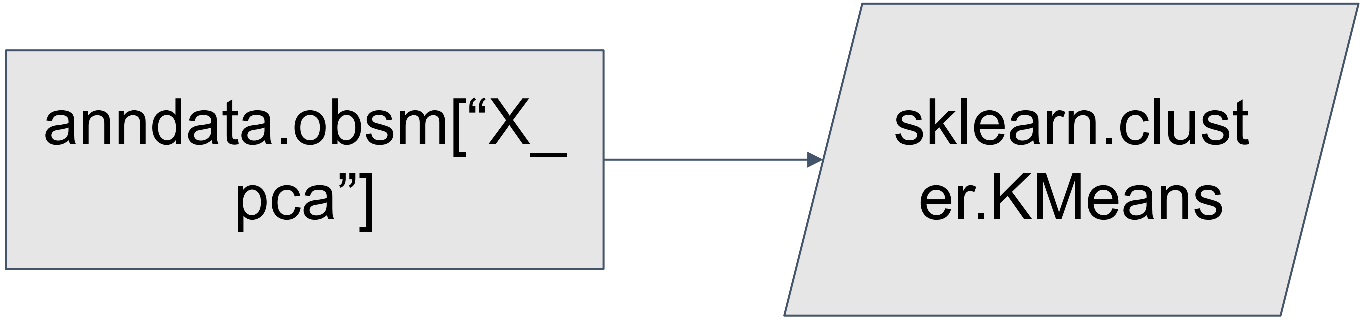
1. Extract the PCs components from AnnData for every cells/spots



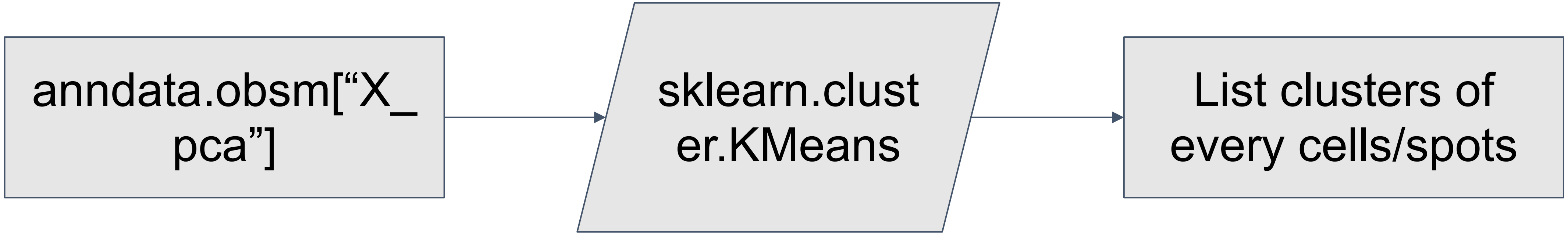
AnnData:
anndata

anndata.obsm[“X_pca”]

2. Using external scikit-learn package for K-means clustering



3. Get the K-means clustering results

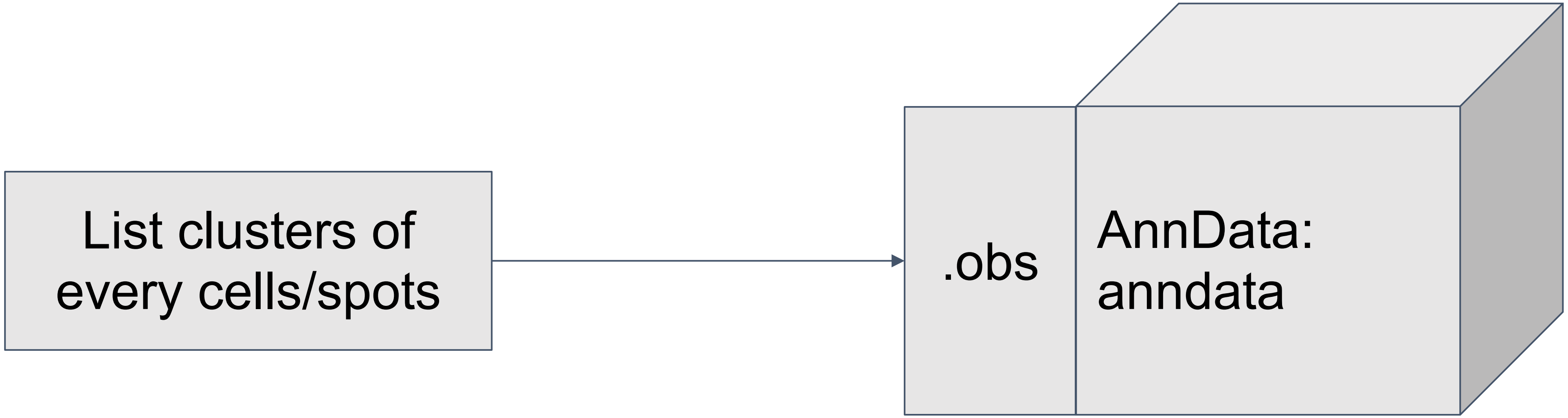


anndata.obsm[“X_pca”]

sklearn.cluster.KMeans

List clusters of every cells/spots

4. Add results to observation annotation of AnnData object

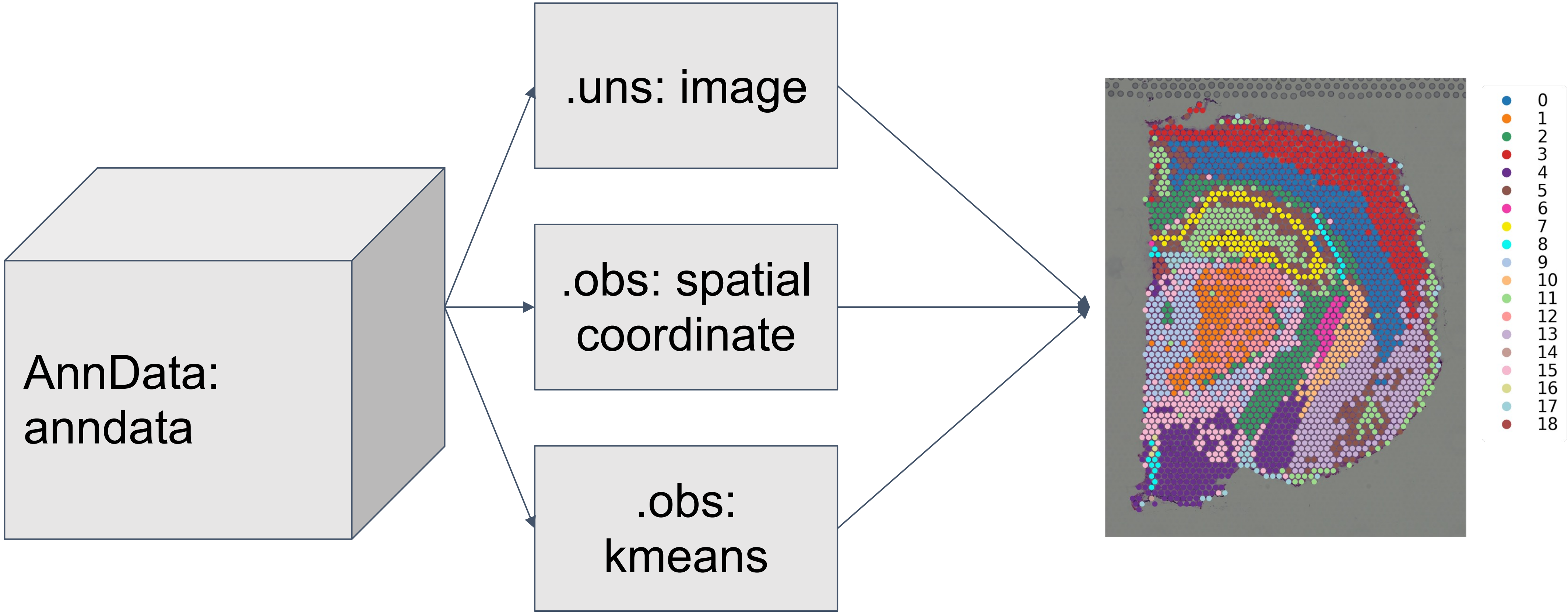


List clusters of every cells/spots

.obs

AnnData: anndata

Use case:
Plotting Kmeans results for spatial transcriptomics



AnnData: anndata

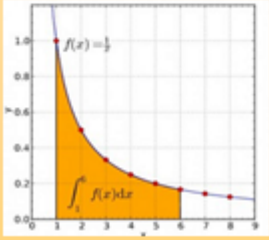
.uns: image

.obs: spatial coordinate

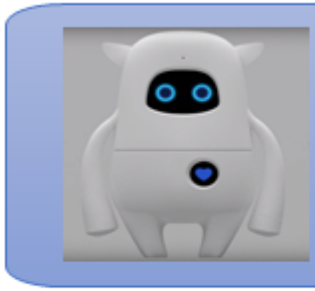
.obs: kmeans

Introduction machine learning

Definition of machine learning is an unsettled topic, but is important to know



Statistics +



Machine Learning =

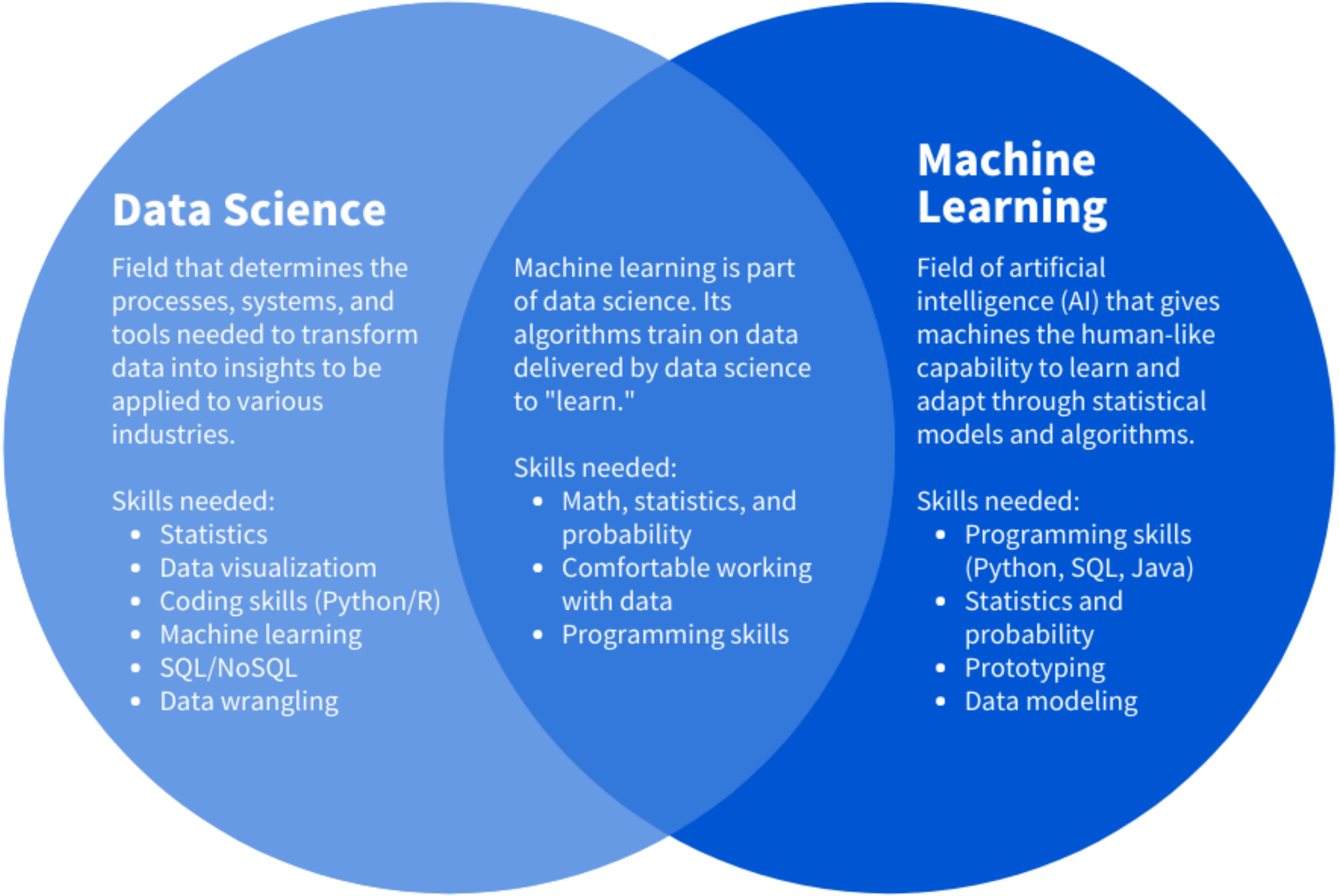
$$\sum_{i=1}^{n}(y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Statistical Learning

	Statistics	Machine Learning	Statistical Learning
Subfield of…	Mathematics	Computer Science (AI)	Statistics & Machine Learning
Focus on…	Building models with explicitly programmed instructions	Creating systems that learn from data	Sets of tools for modeling and understanding complex data
Purpose	Inferences; Relationships between variables	Optimization; Prediction accuracy	Building statistical models for prediction; understanding data
Prior assumptions about data	Some knowledge about population usually required	None	Some knowledge about population may be required
Dimensionality of data	Usually applied to low-dimensional data	Usually applied to high dimensional data; ML learns from data	Usually applied to high dimensional data
Knowledge overlap	No ML knowledge required	Some stats knowledge usually needed: stats is basis for algorithms	Knowledge of statistics and ML required

Musio image: Akawikipic [CC BY-SA 4.0 (https://creativecommons.org/licenses/by-sa/4.0)]

Take home message: ML and SL are essentially the same; recent trends see the increased used of statistics in ML



Data Science

Field that determines the processes, systems, and tools needed to transform data into insights to be applied to various industries.

Skills needed:

- Statistics
- Data visualizatiom
- Coding skills (Python/R)
- Machine learning
- SQL/NoSQL
- Data wrangling

Machine learning is part of data science. Its algorithms train on data delivered by data science to "learn."

Skills needed:

- Math, statistics, and probability
- Comfortable working with data
- Programming skills

Machine Learning

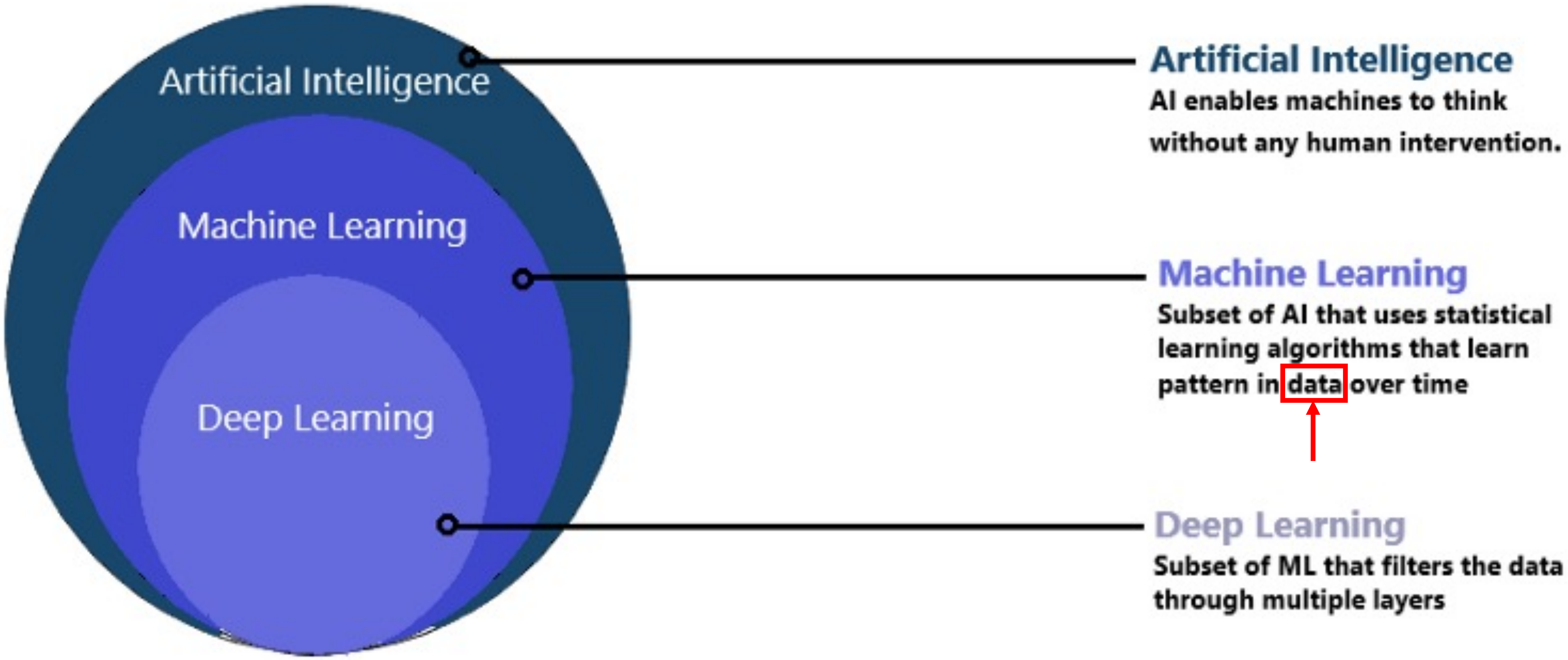
Field of artificial intelligence (AI) that gives machines the human-like capability to learn and adapt through statistical models and algorithms.

Skills needed:

- Programming skills (Python, SQL, Java)
- Statistics and probability
- Prototyping
- Data modeling

(Coursera, 2022)

Machine learning, statistical learning, deep learning



Artificial Intelligence

AI enables machines to think without any human intervention.

Machine Learning

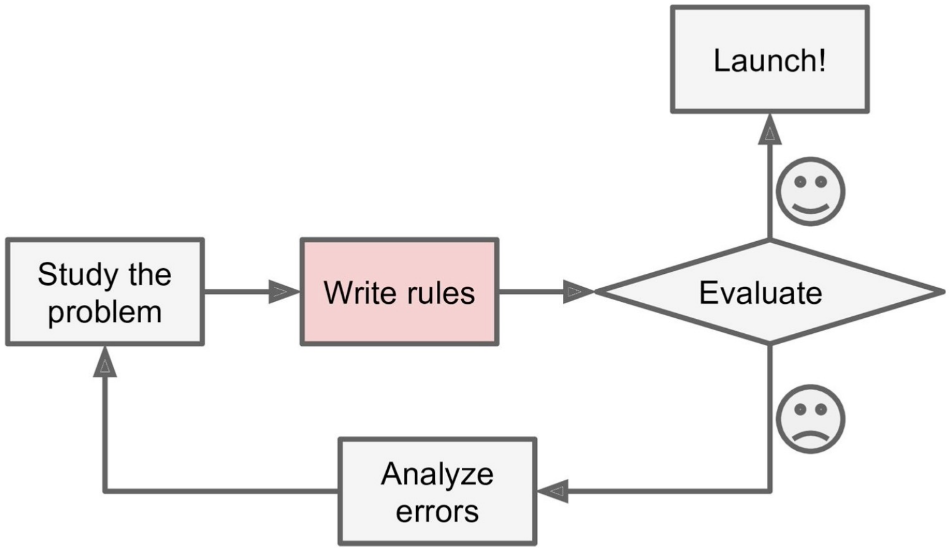
Subset of AI that uses statistical learning algorithms that learn pattern in data over time

Deep Learning

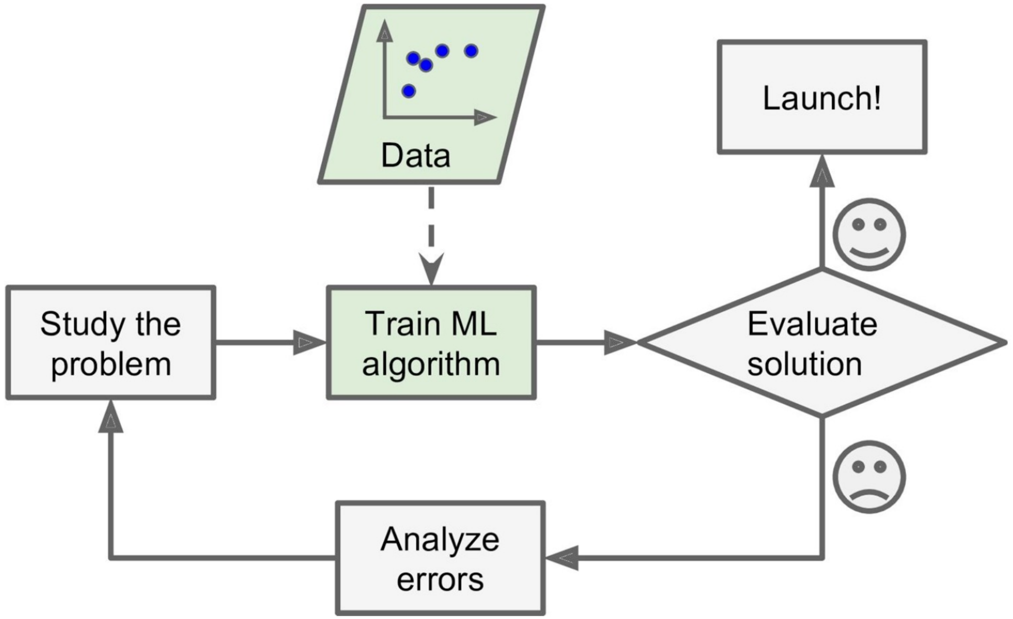
Subset of ML that filters the data through multiple layers

Machine learning vs programming

- The training of programs developed by allowing a computer to learn from its experience (rather than through manually coding the individual steps)
- A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E (Tom Mitchell, 1997)

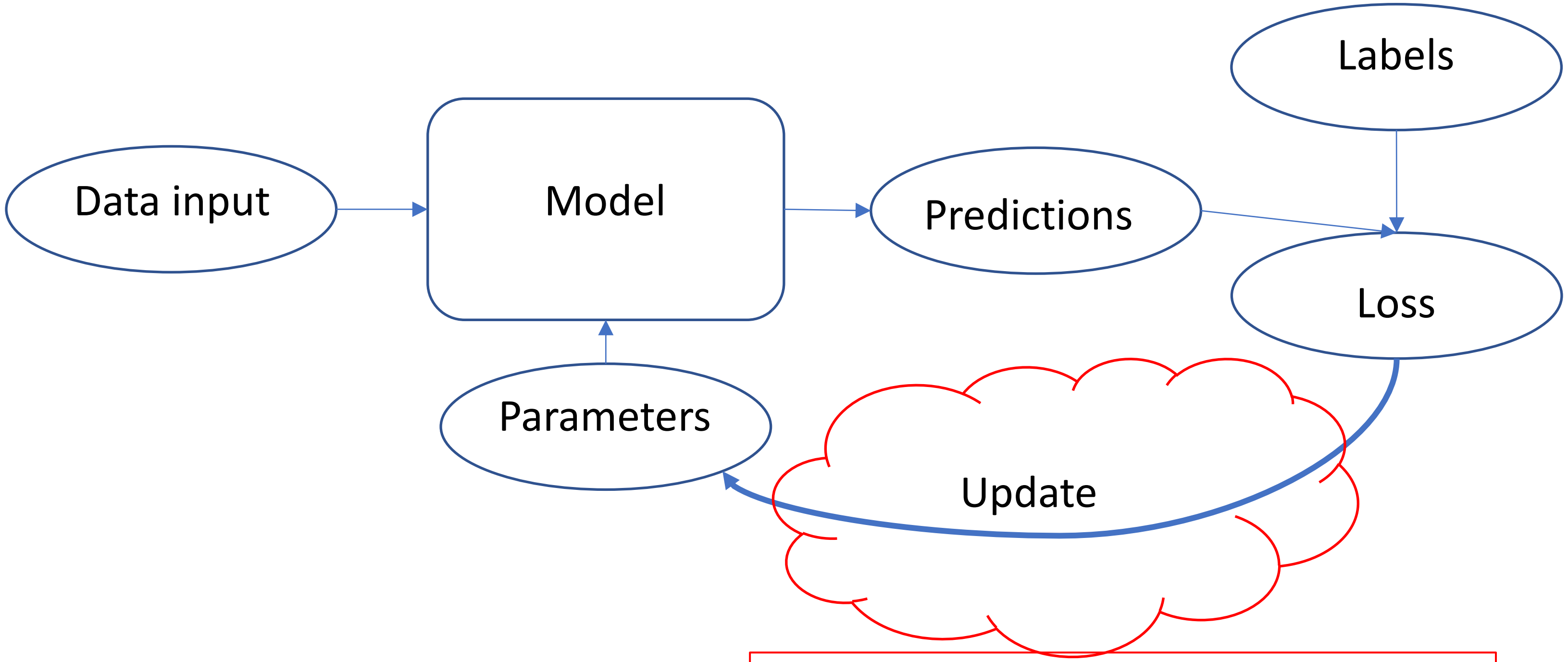


Programming



Machine learning

Machine learning – Loss function



Labels

Data input

Model

Predictions

Loss

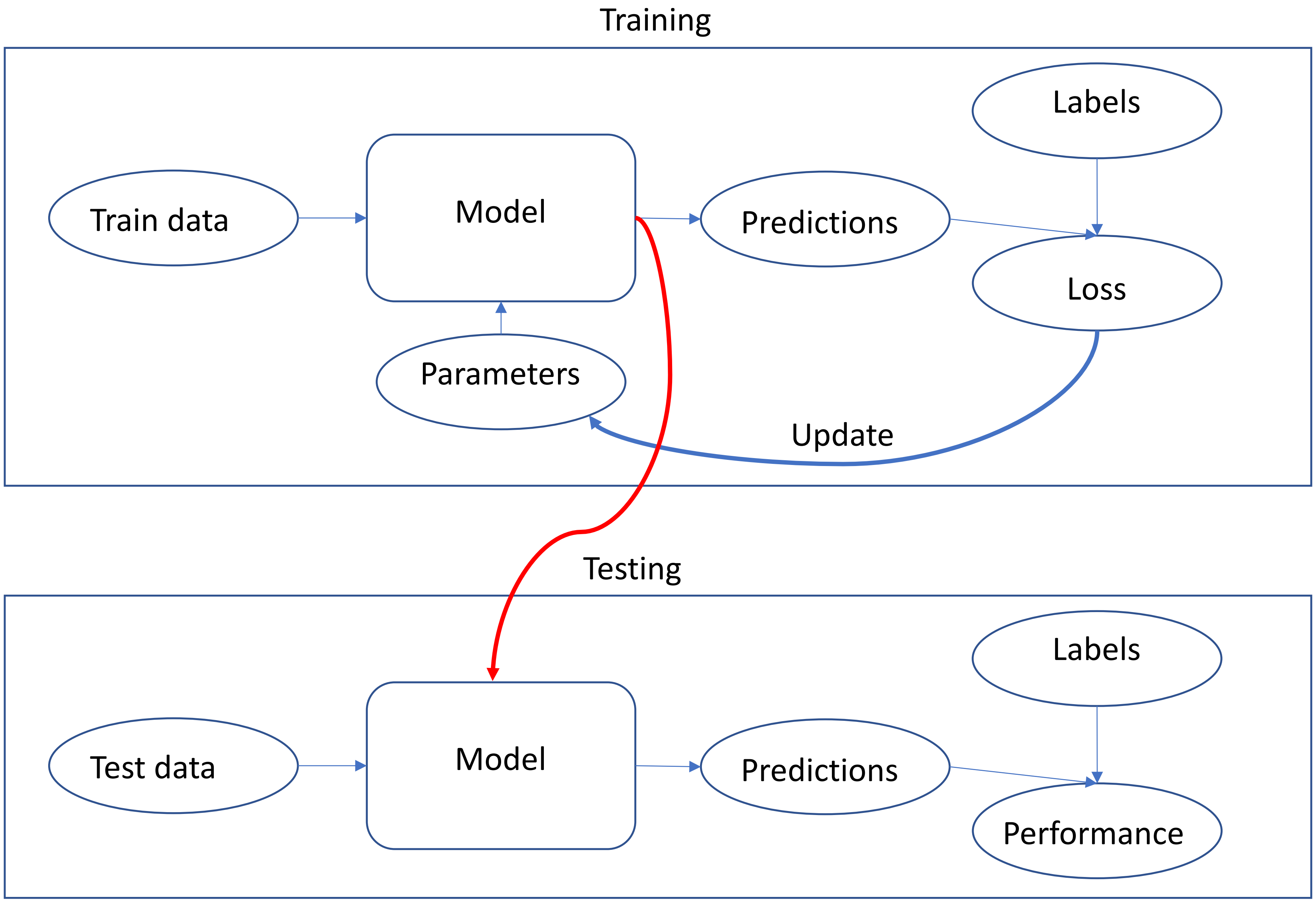
Parameters

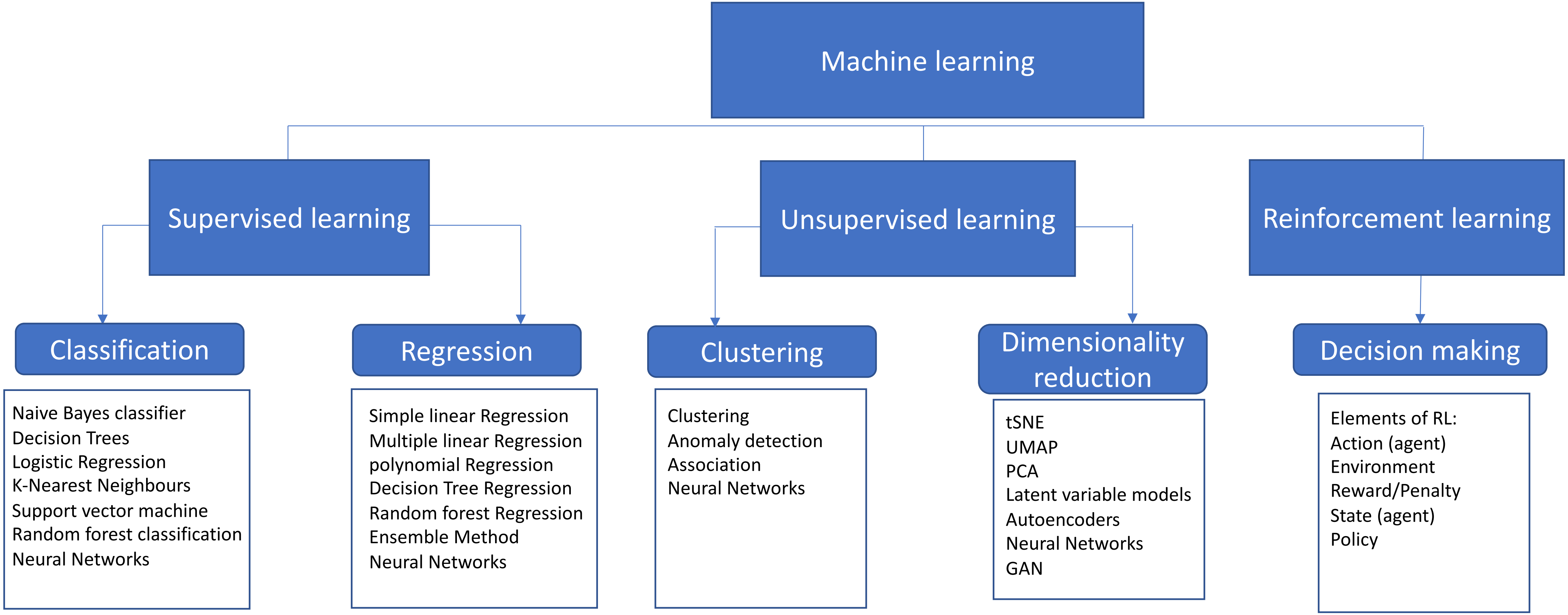
Update

This is where the machine is learning

- ML: The training of programs developed by allowing a computer to learn from its experience (rather than through manually coding the individual steps)
- Loss function is where ML meets statistical models
- (hyper)Parameters are where machine learning deviate from statistical models

Machine learning – Training and testing datasets





Machine learning

Supervised learning

Unsupervised learning

Reinforcement learning

Classification

Naive Bayes classifier
Decision Trees
Logistic Regression
K-Nearest Neighbours
Support vector machine
Random forest classification
Neural Networks

Regression

Simple linear Regression
Multiple linear Regression
polynomial Regression
Decision Tree Regression
Random forest Regression
Ensemble Method
Neural Networks

Clustering

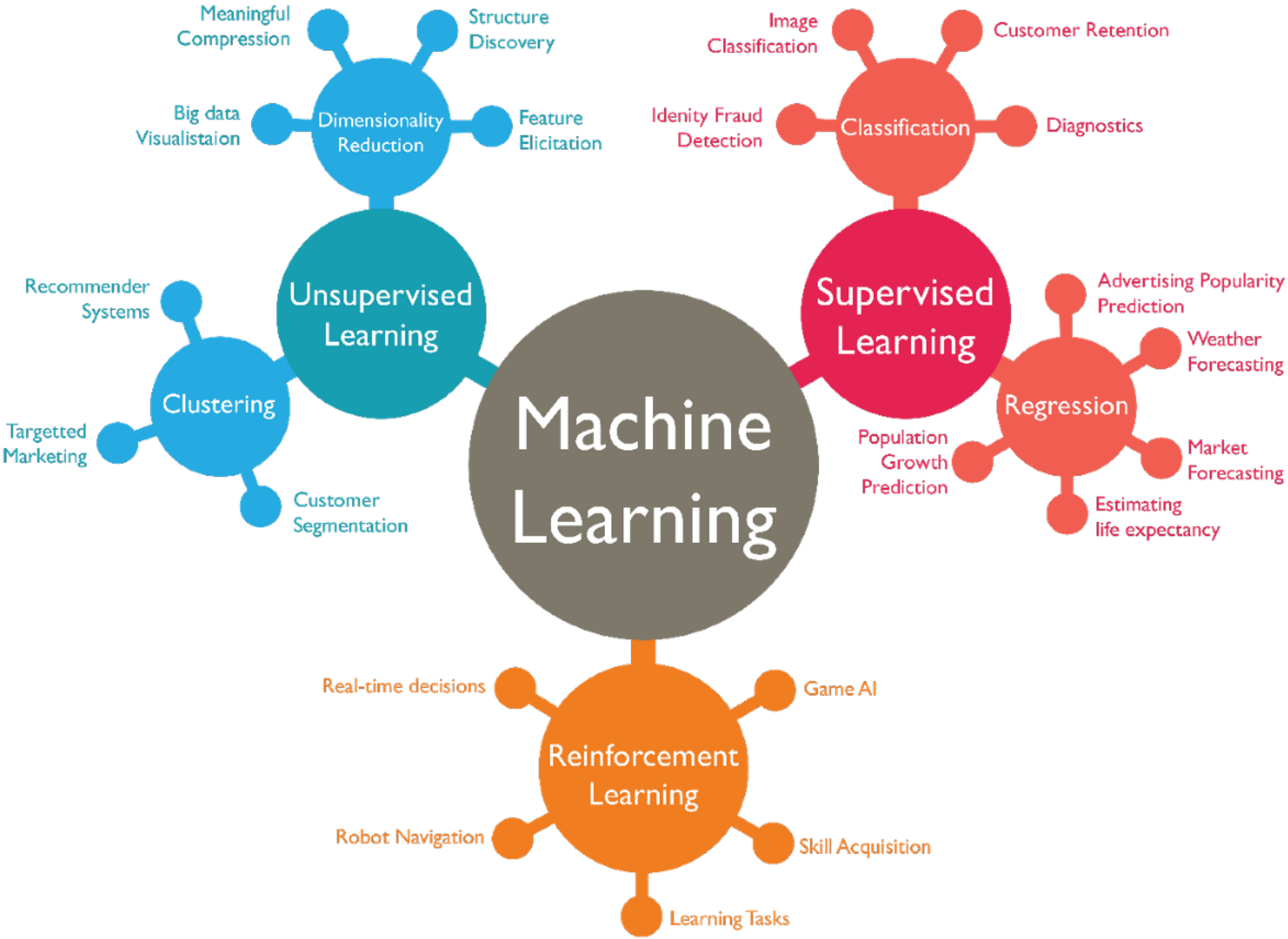
Clustering
Anomaly detection
Association
Neural Networks

Dimensionality reduction

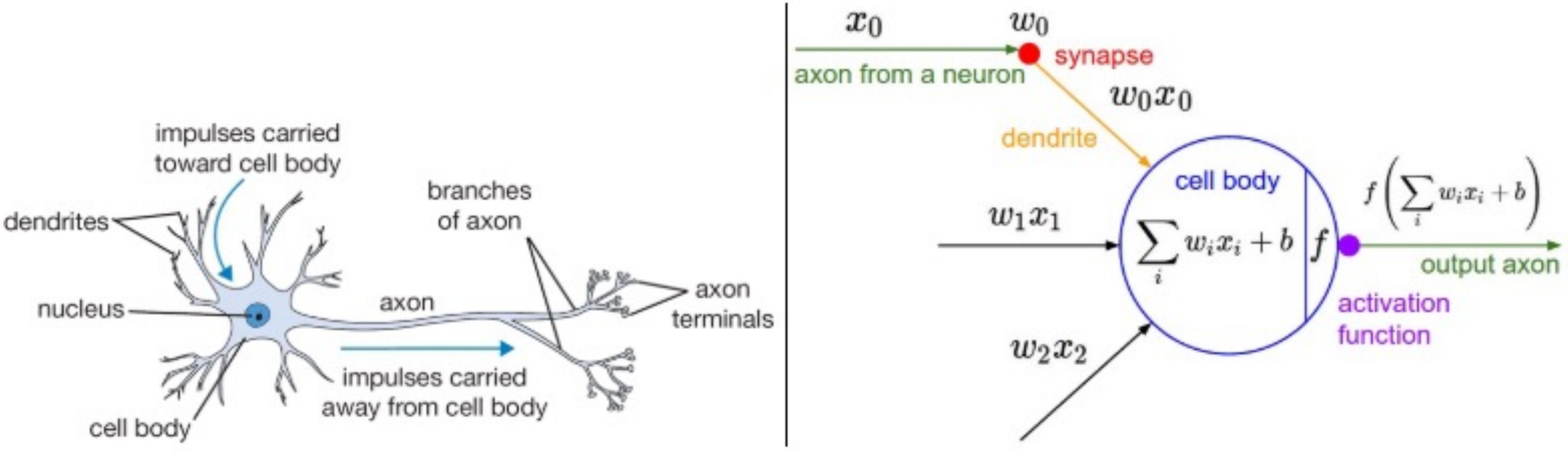
tSNE
UMAP
PCA
Latent variable models
Autoencoders
Neural Networks
GAN

Decision making

Elements of RL:
Action (agent)
Environment
Reward/Penalty
State (agent)
Policy


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

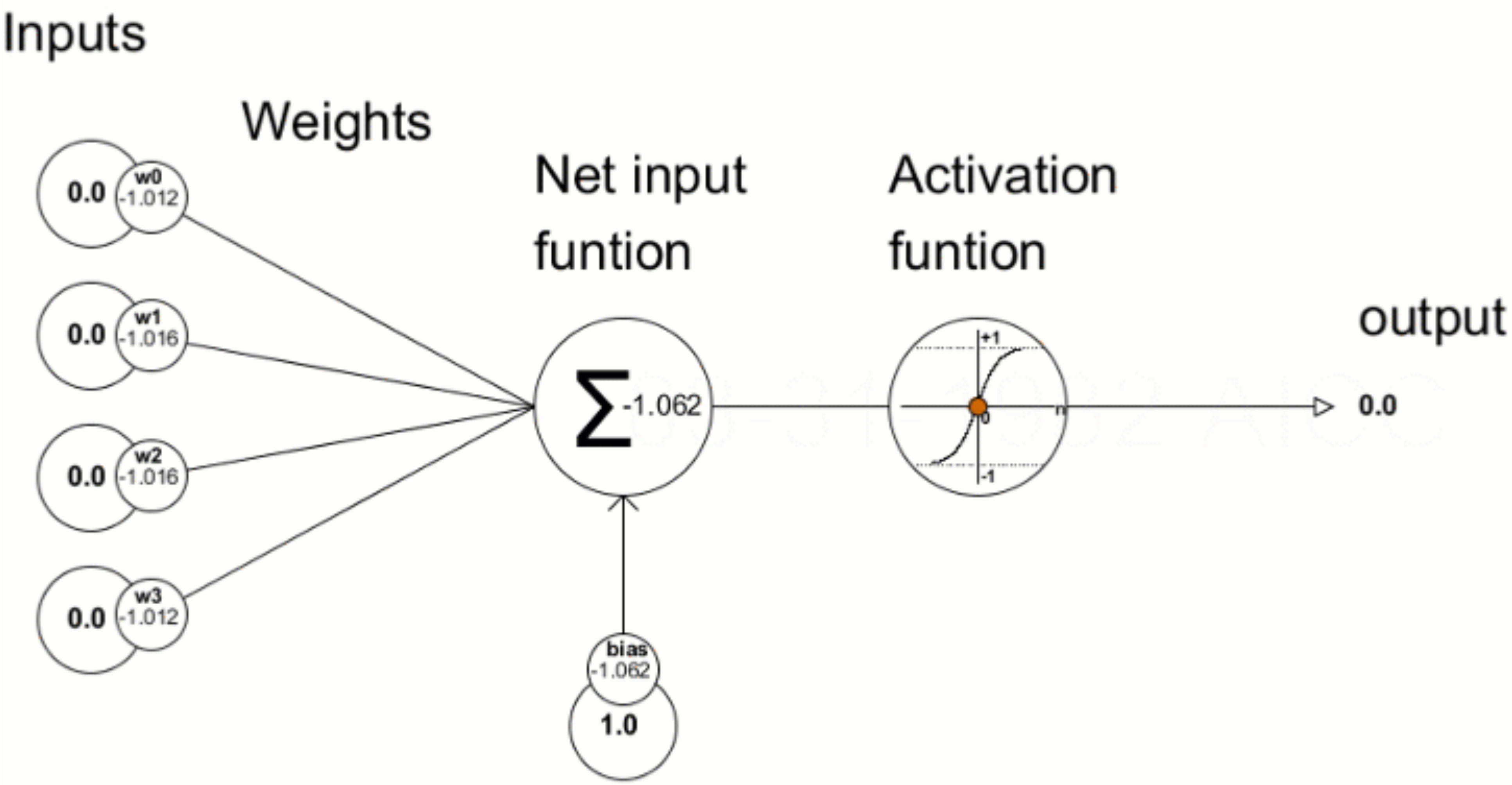
Deep learning – Neural network



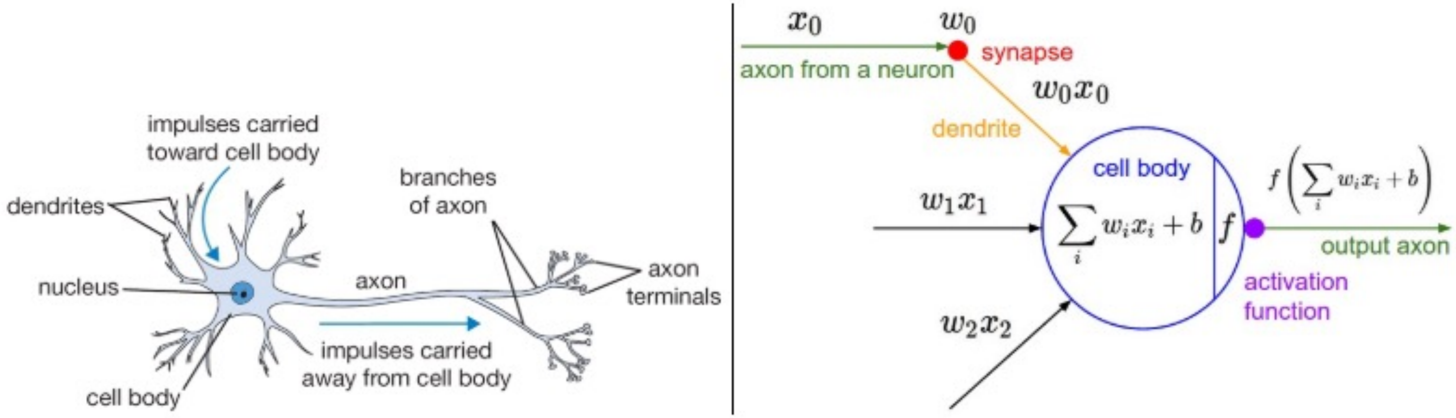
(Source: cs231n, Stanford)

$$Y = \sum (weight * input) + bias$$

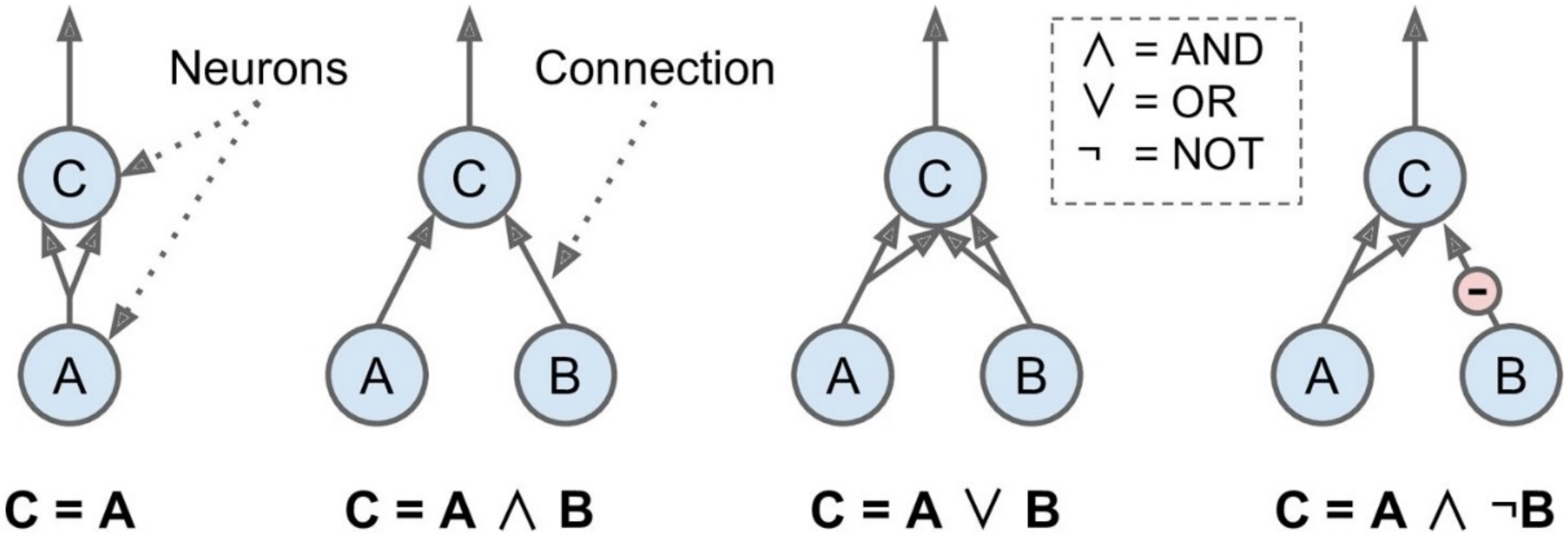
Single neuron in action – activation function



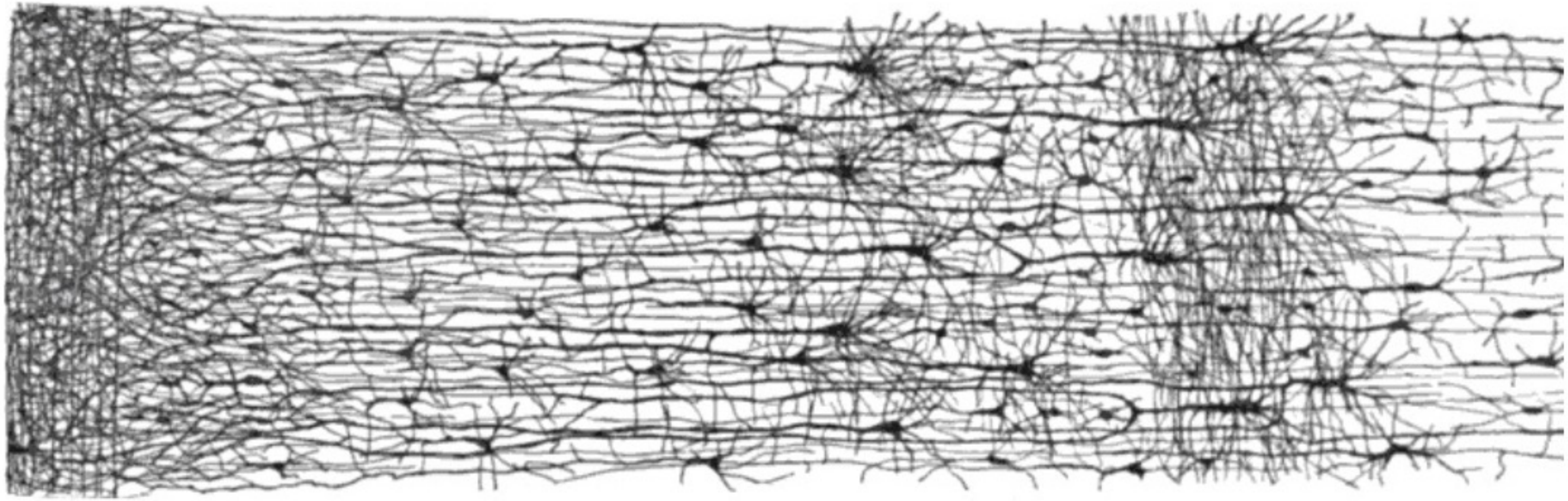
(Towards AI, 2019)

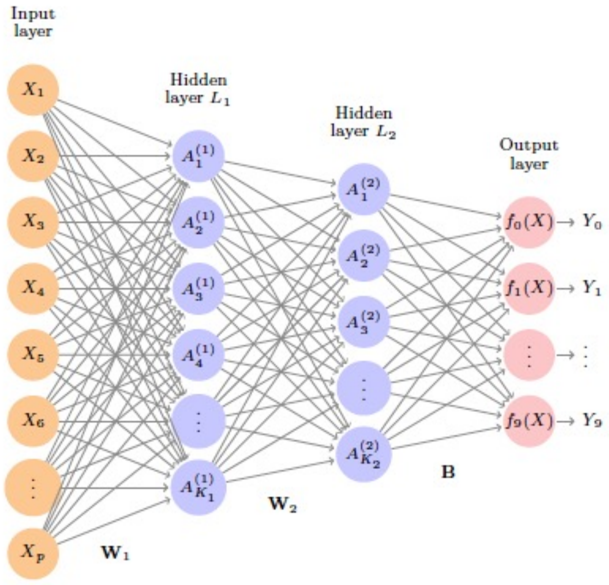


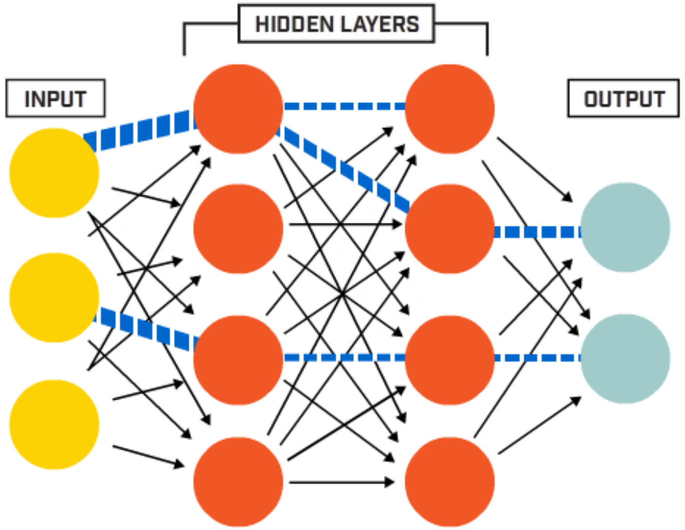
(Source: cs231n, Stanford)



Multilayer perceptrons

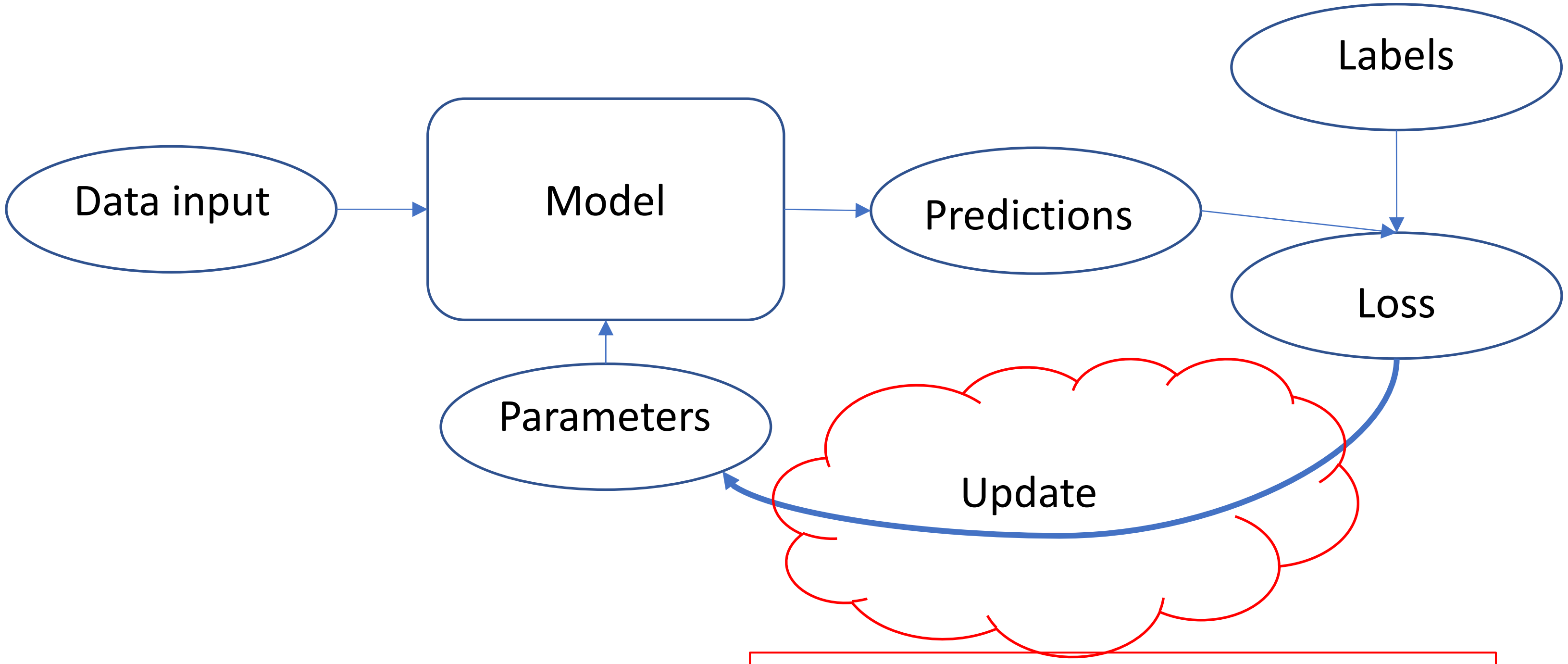






(Towards AI, 2019)

Machine learning – Loss function



Labels

Data input

Model

Predictions

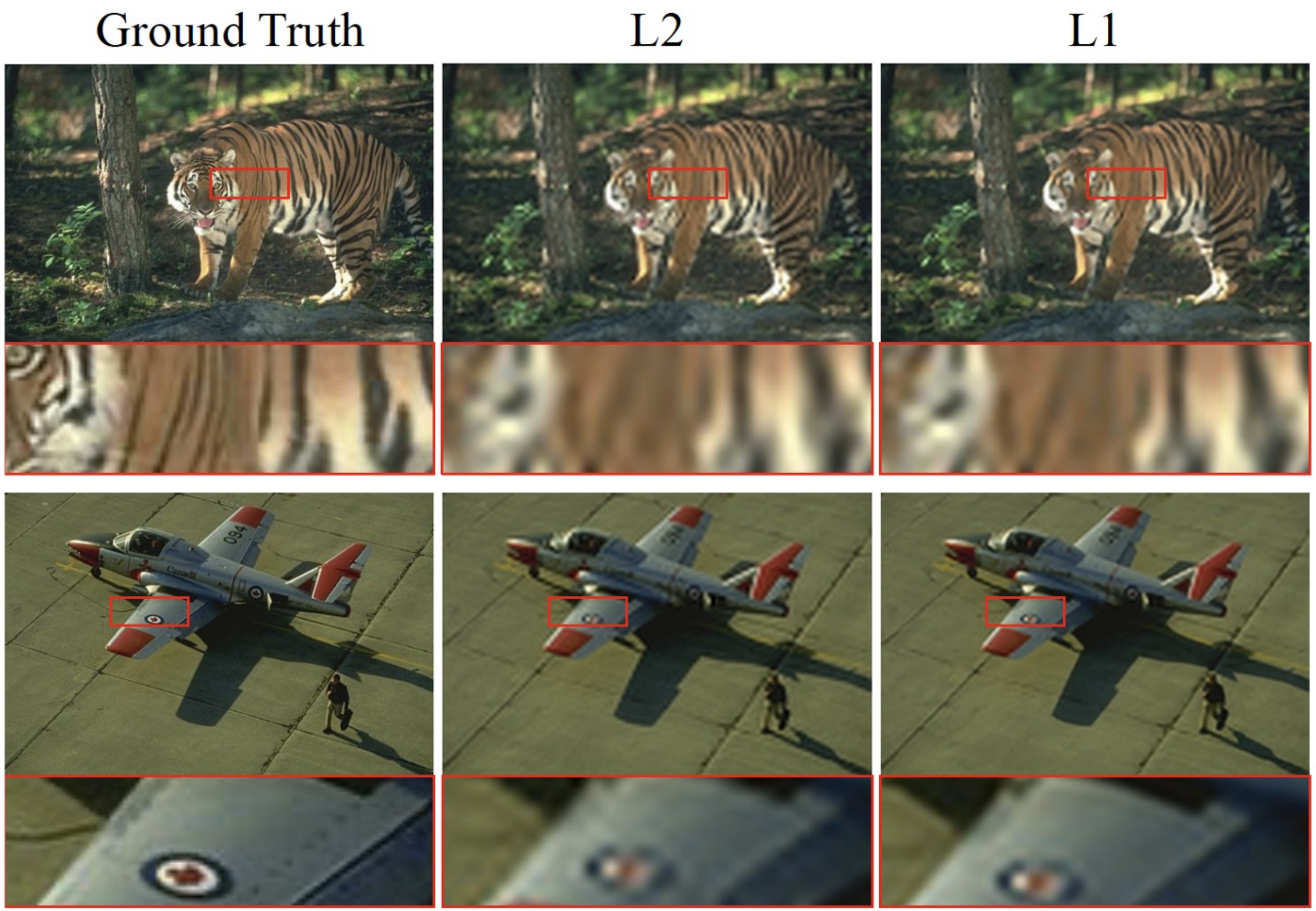
Loss

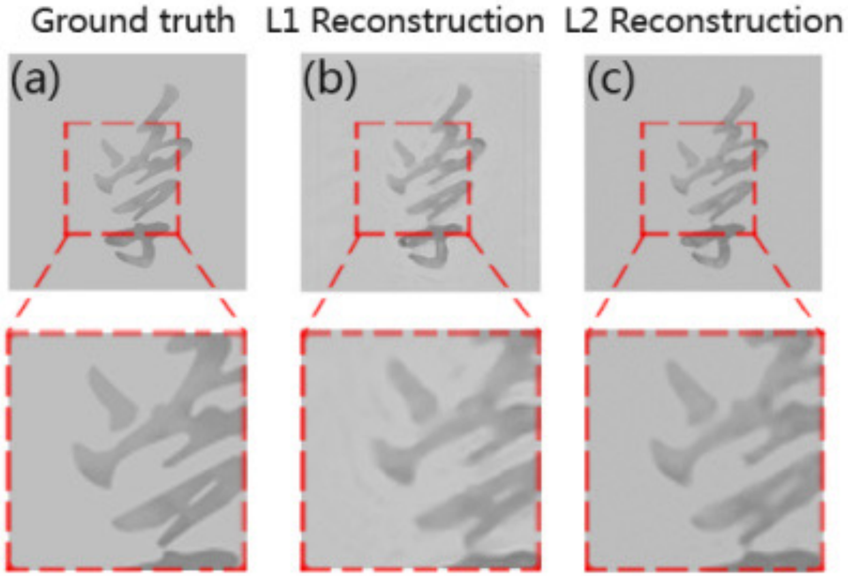
Parameters

Update

This is where the machine is learning

Pixel-wise loss function



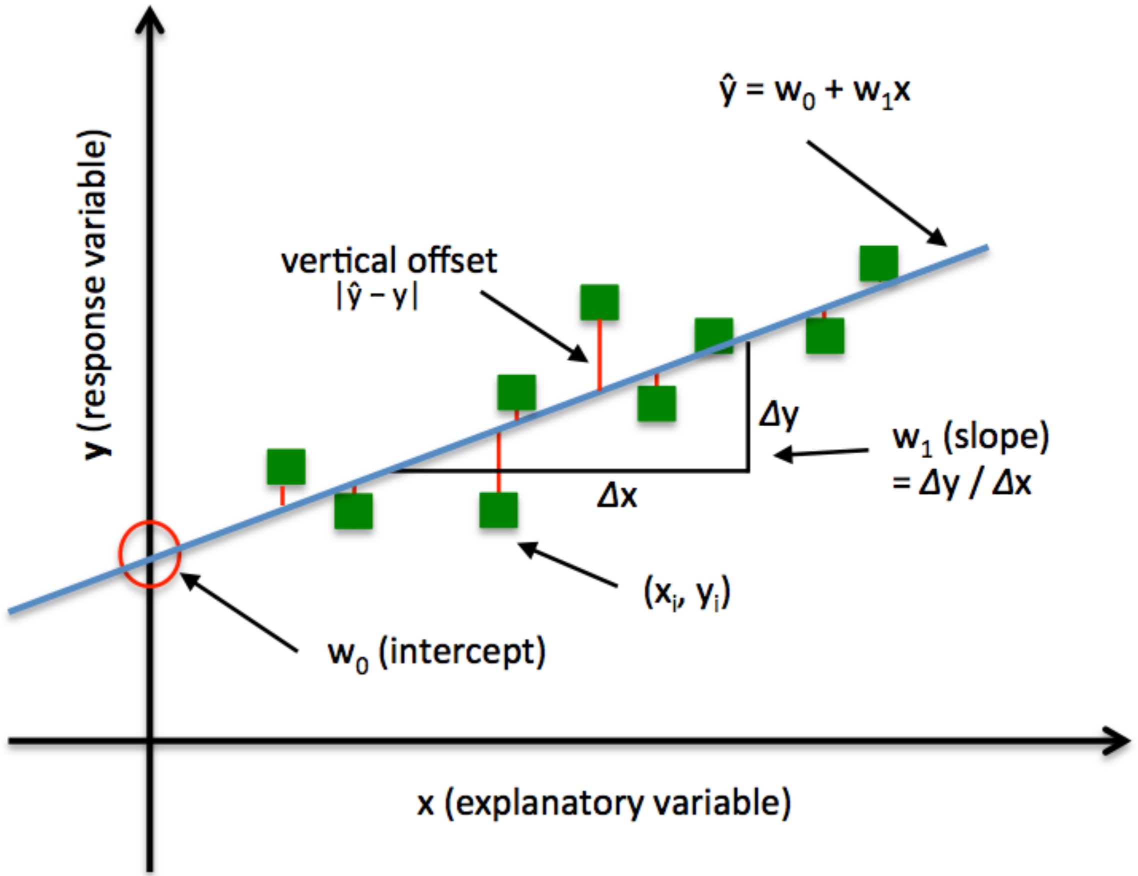


$$L2LossFunction = \sum_{i=1}^{n}(y_{true} - y_{predicted})^2$$

$$L1LossFunction = \sum_{i=1}^{n}|y_{true} - y_{predicted}|$$

Introduction to machine learning:
key concepts and a few classical ML models

General terms exampled by regressions



$$Error = \frac{1}{N}\sum_{i=1}^{N}(y_i \;- \hat{y}_i)^2$$

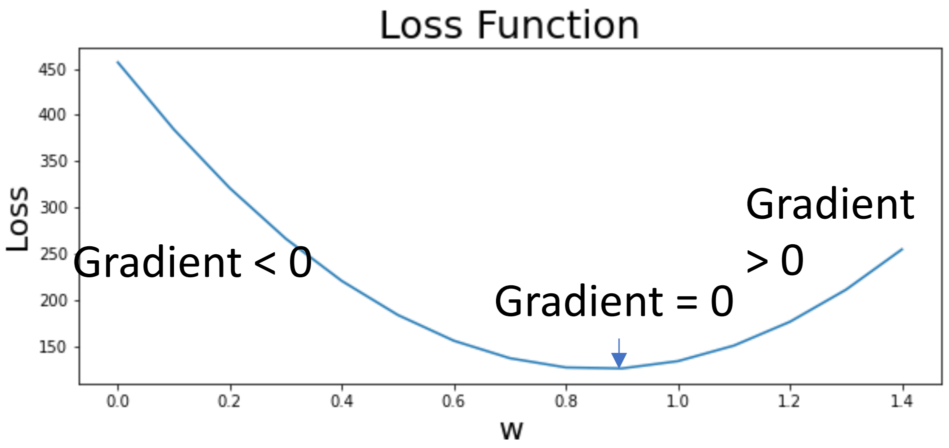
$$= \frac{1}{N}\sum_{i=1}^{N}(y_i \;- w_0 - w_1 X_i \;)^2$$

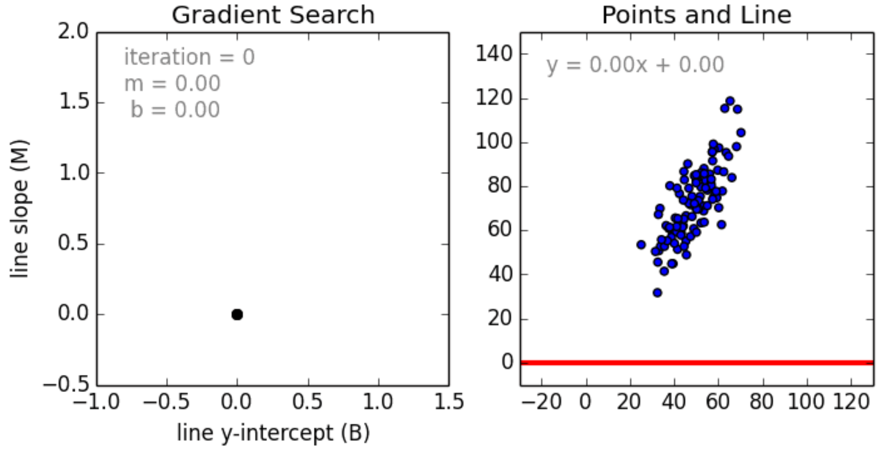
= Objective function
= Loss function
= J(w_0, w_1)

To minimize wrt w_0 and w_1 by gradient descent

http://rasbt.github.io/mlxtend/user_guide/regressor/LinearRegression/

General terms: Gradient Descent Example for Linear Regression

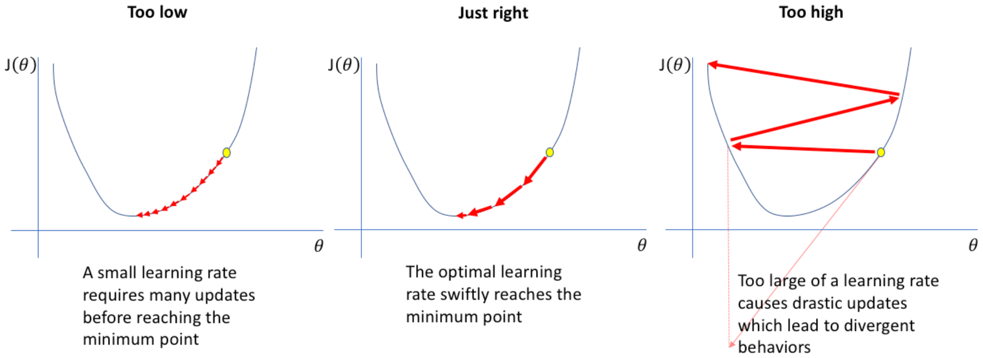




$$w^t = w^{t-1} - \alpha \nabla J(w^{t-1})$$

α is the learning rate (step length)

Effect of learning rate →



A small learning rate requires many updates before reaching the minimum point

The optimal learning rate swiftly reaches the minimum point

Too large of a learning rate causes drastic updates which lead to divergent behaviors

https://www.jeremyjordan.me/nn-learning-rate/

https://github.com/mattnedrich/GradientDescentExample

General terms: often used different loss functions

Regression:

Mean Square Error/Quadratic Loss/L2 Loss: $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i \ - \hat{y_i})^2$

Mean Absolute Error/L1 Loss: $MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i \ - \hat{y_i}|$

Mean Bias Error: $MBE = \frac{1}{N}\sum_{i=1}^{N}(y_i \ - \hat{y_i})$

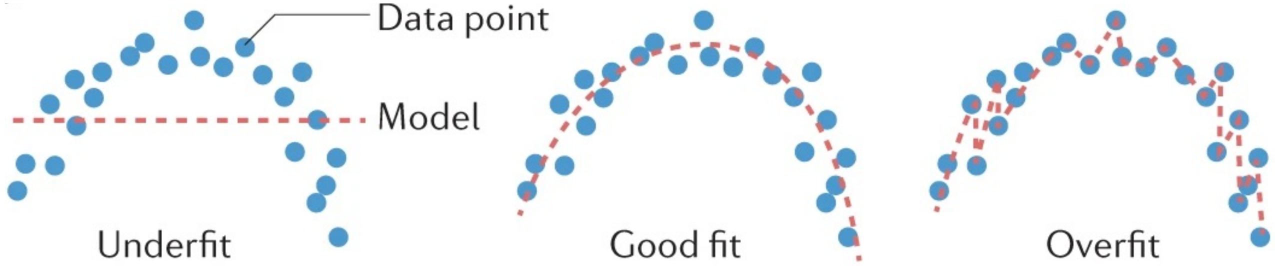
Negative Log Likelihood

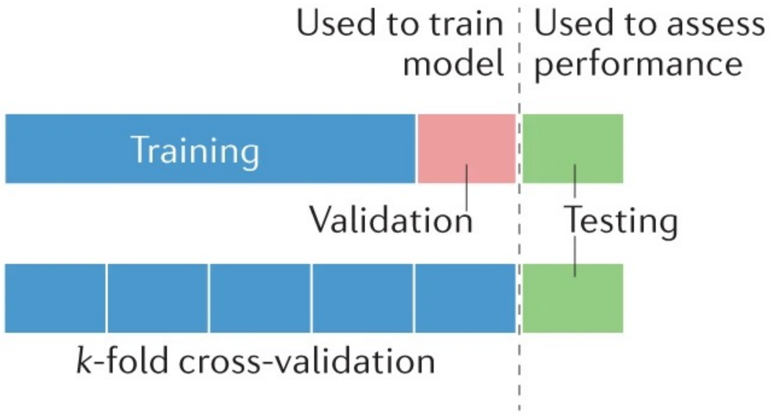
Classification:

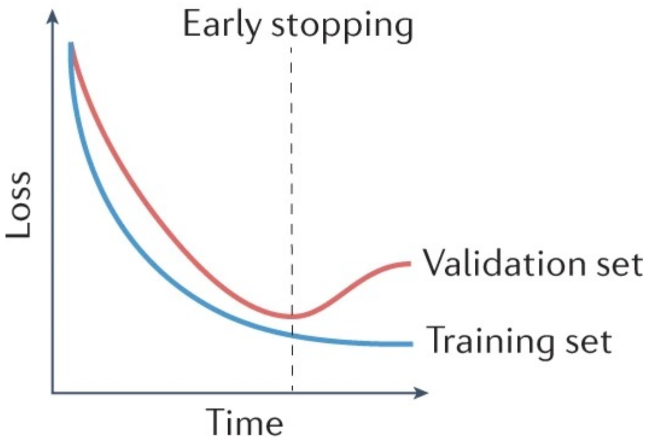
Cross Entropy Loss/Negative Log Likelihood: $-(y_i \ \log(\hat{y_i}) + (1 - y_i \ ) \log(\widehat{1 - y_i}))$

https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23

General terms : Overfitting and how to reduce

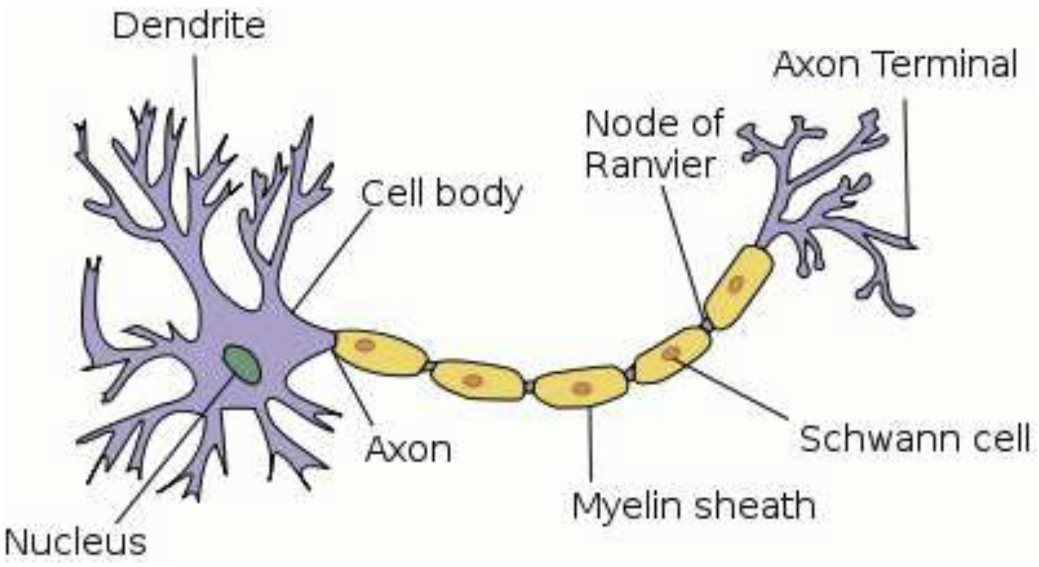






Neural network methods

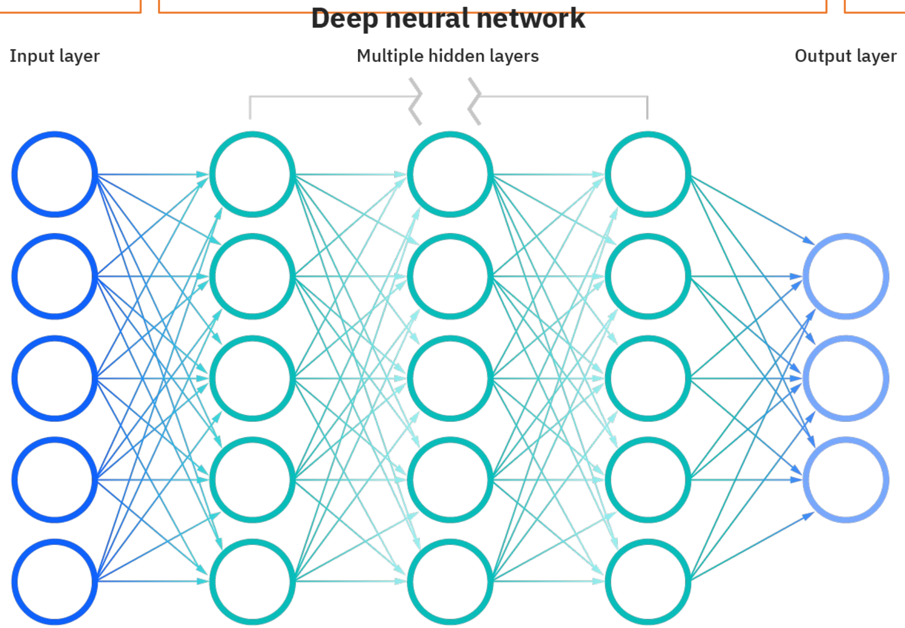
Inspired by Neurons



(lots) Inputs

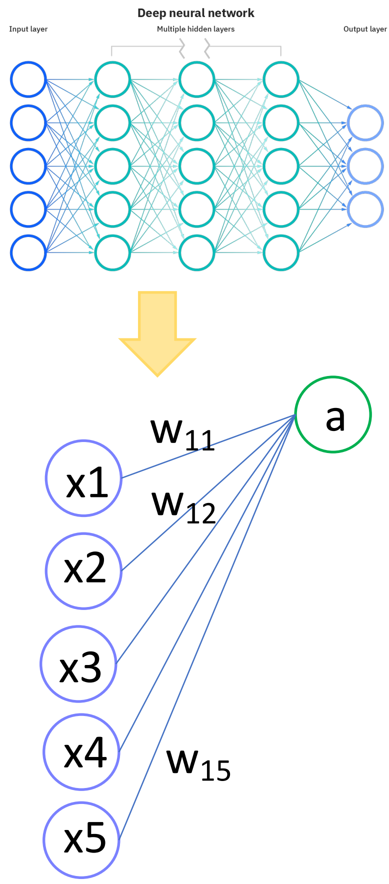
Process the signal from inputs

Output results

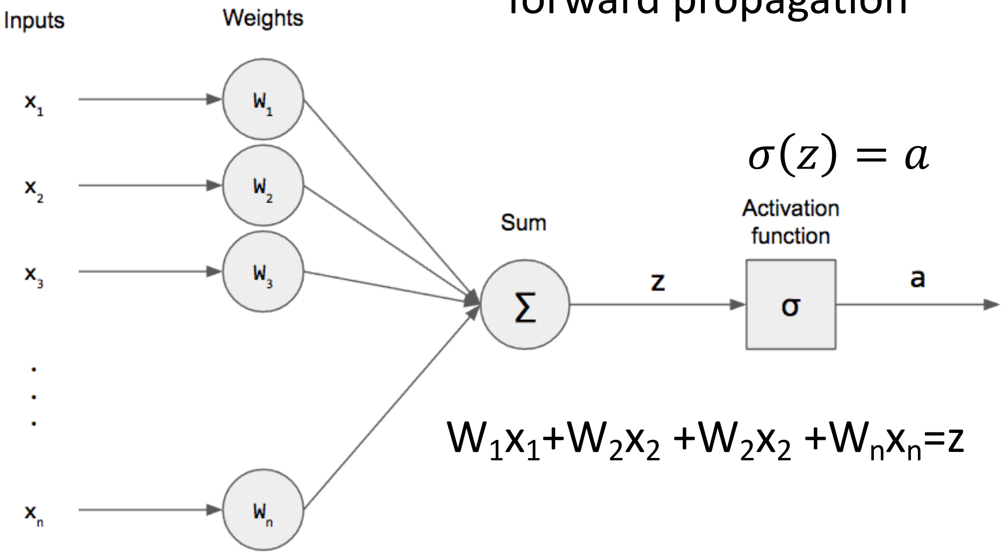


https://www.ibm.com/cloud/learn/neural-networks

Multilayer perceptron – foundation of other neural networks



forward propagation

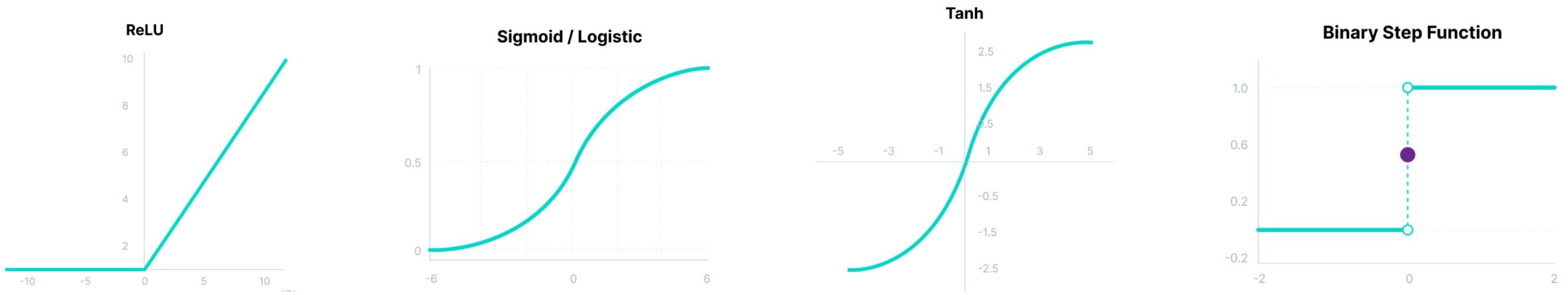


$$\sigma(z) = a$$

Next layer

$$W_1x_1+W_2x_2 +W_2x_2 +W_nx_n=z$$

Often used activation functions



https://www.v7labs.com/blog/neural-networks-activation-functions

https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

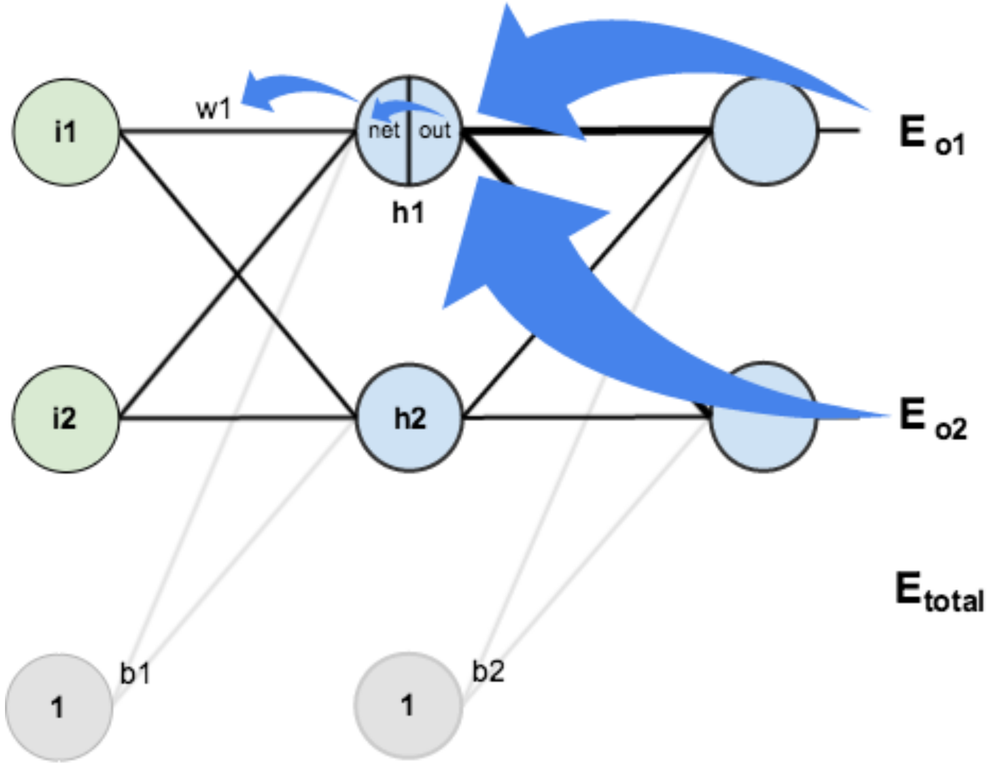
Multilayer perceptron – backward propagation

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\downarrow$$

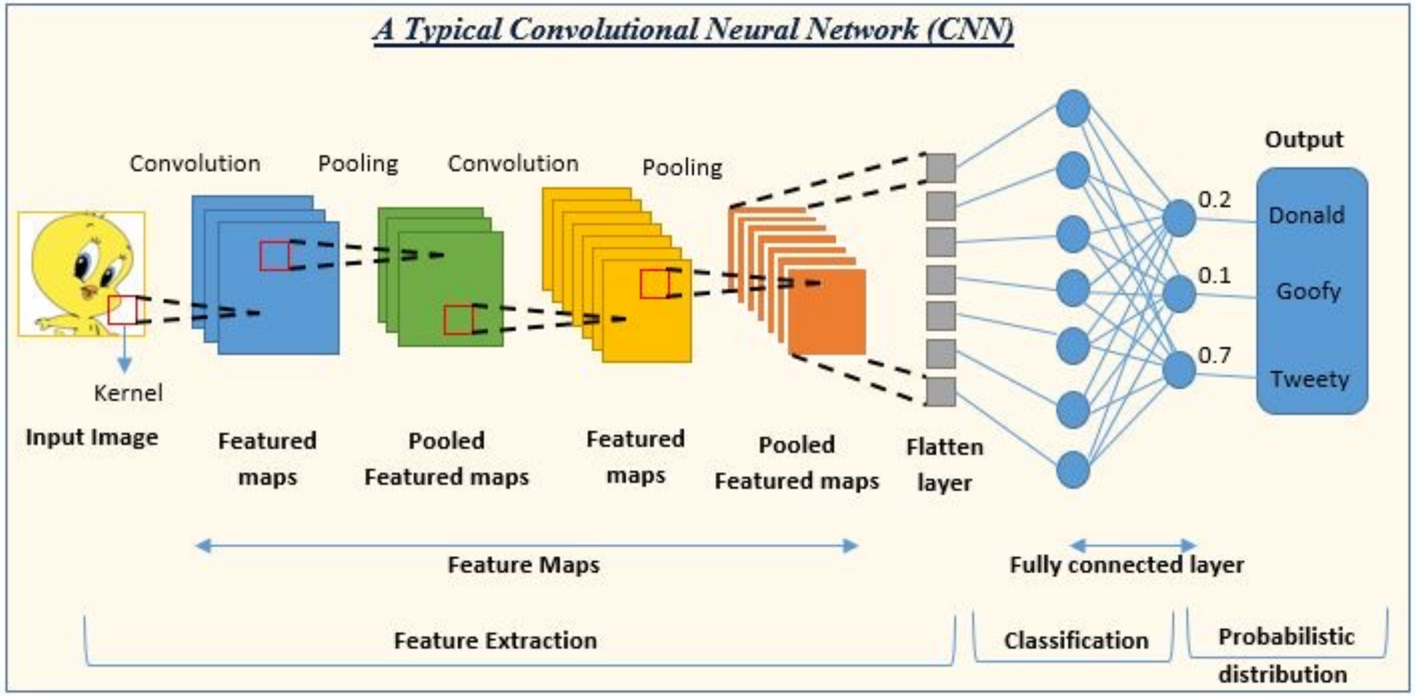
$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$

Chain rule

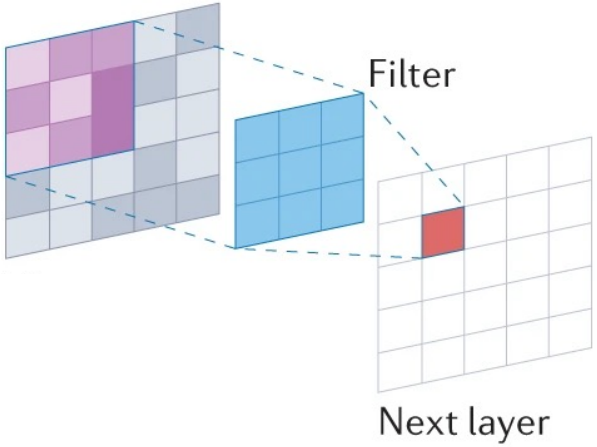


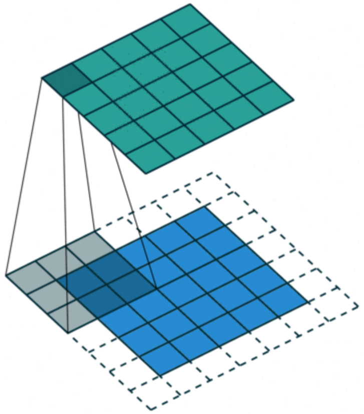
$E_{total} = E_{o1} + E_{o2}$

CNN: convolutional neural network



Convolution

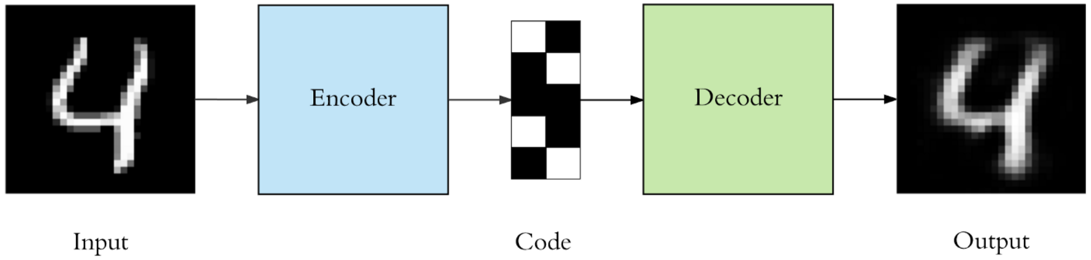


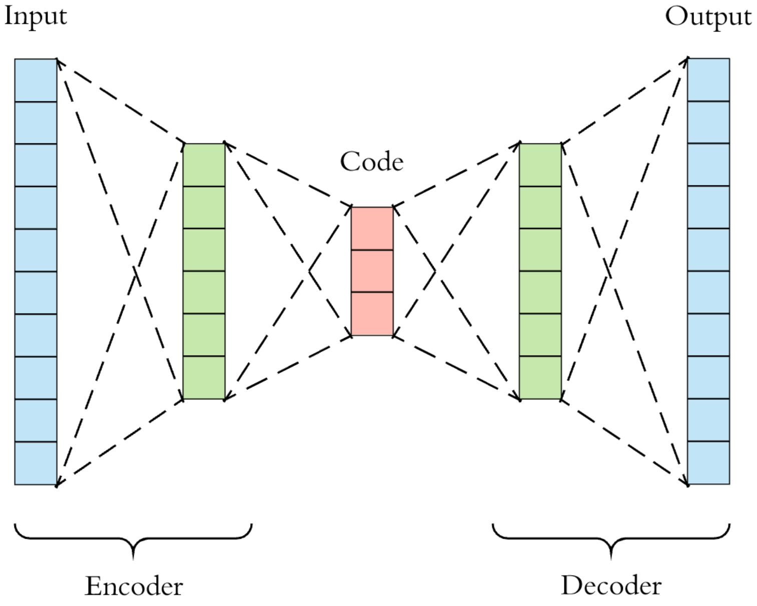


https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/
https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

More networks

Autoencoder





Input is the same as the output compress the input into a lower-dimensional *code (latent-space representation)*

The latent space is determinate

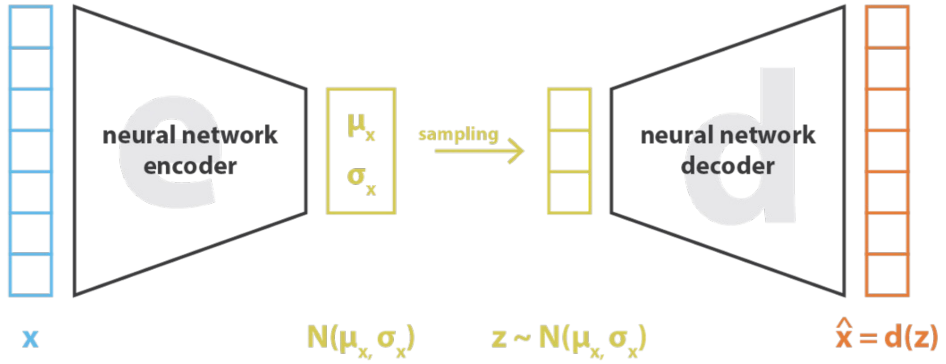
Loss function: KL divergence

https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

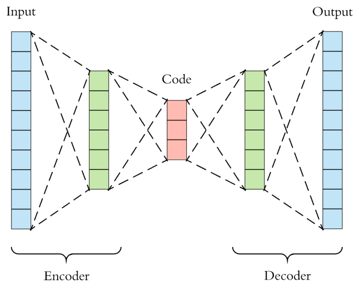
More networks

Variational autoencoder

Laten space becomes distributions



$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,] = \| x - d(z) \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,]$$

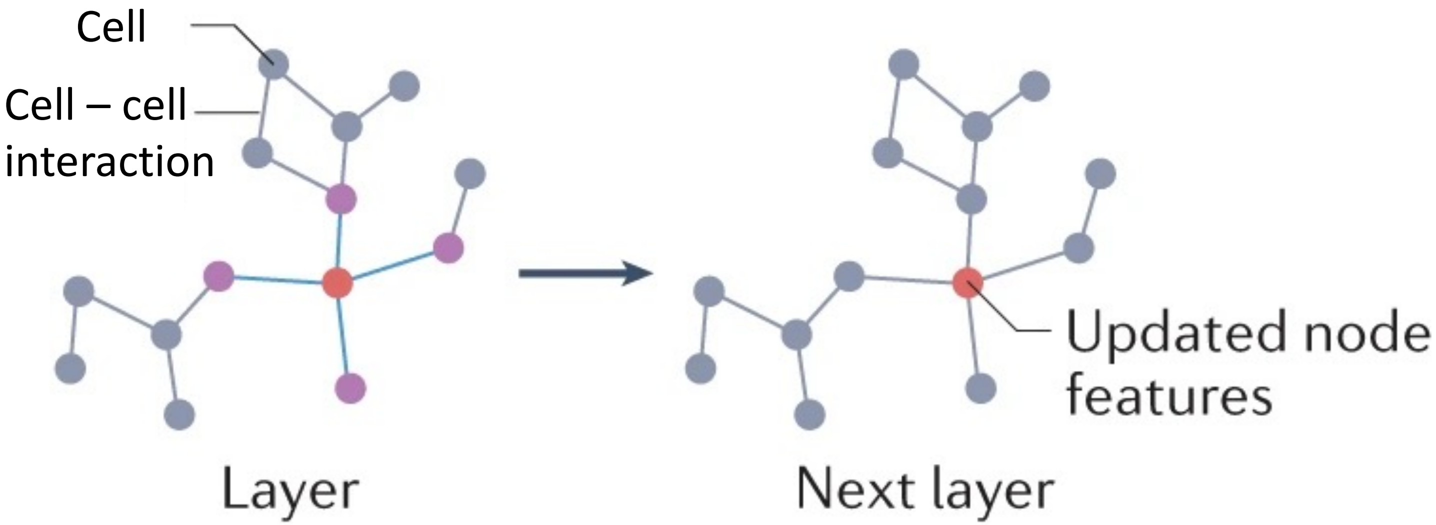


autoencoder

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

More networks

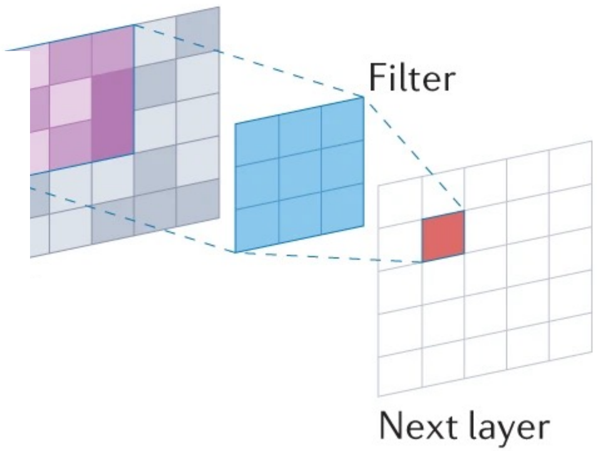
Graph convolutional network



Cell

Cell – cell interaction

Convolution

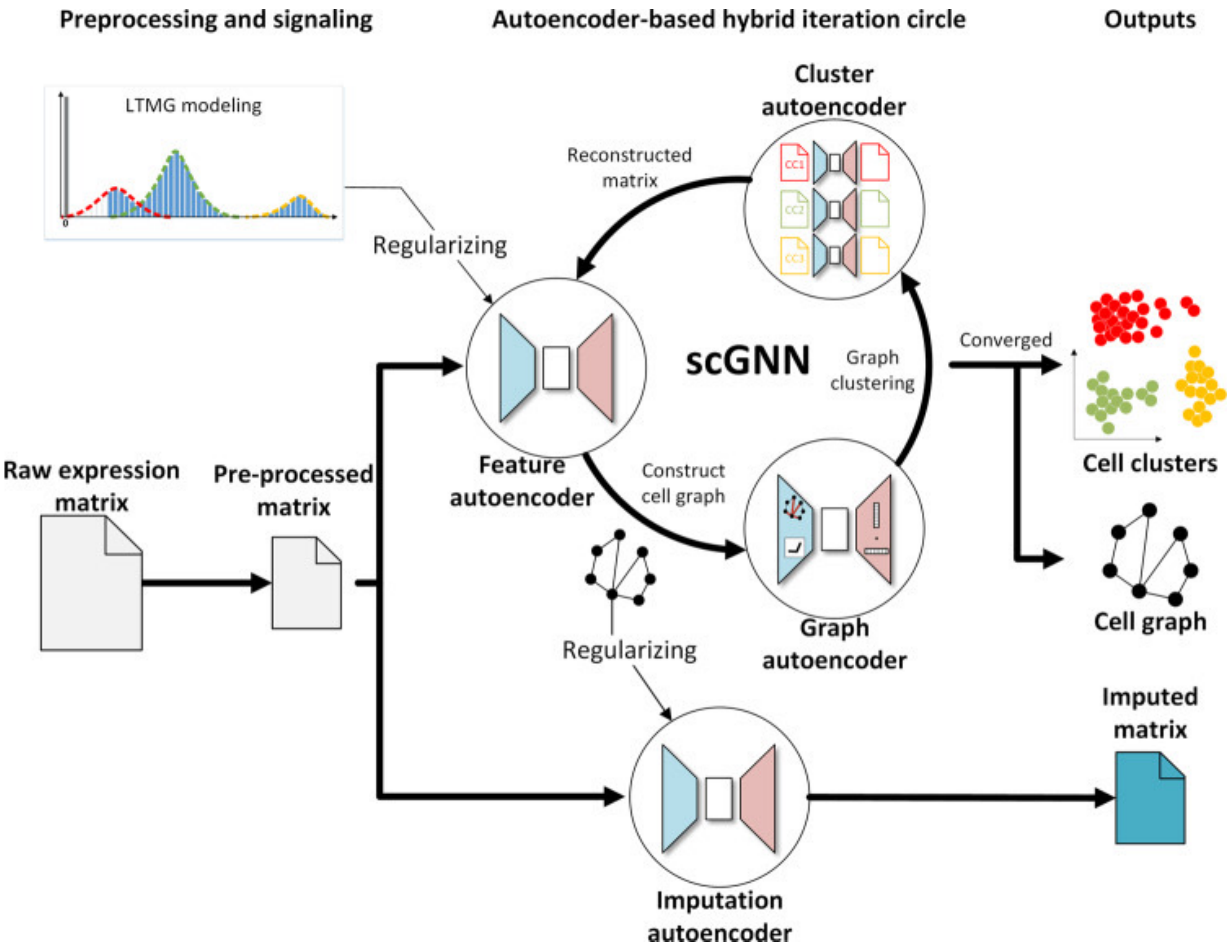


https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Machine learning in single cell data

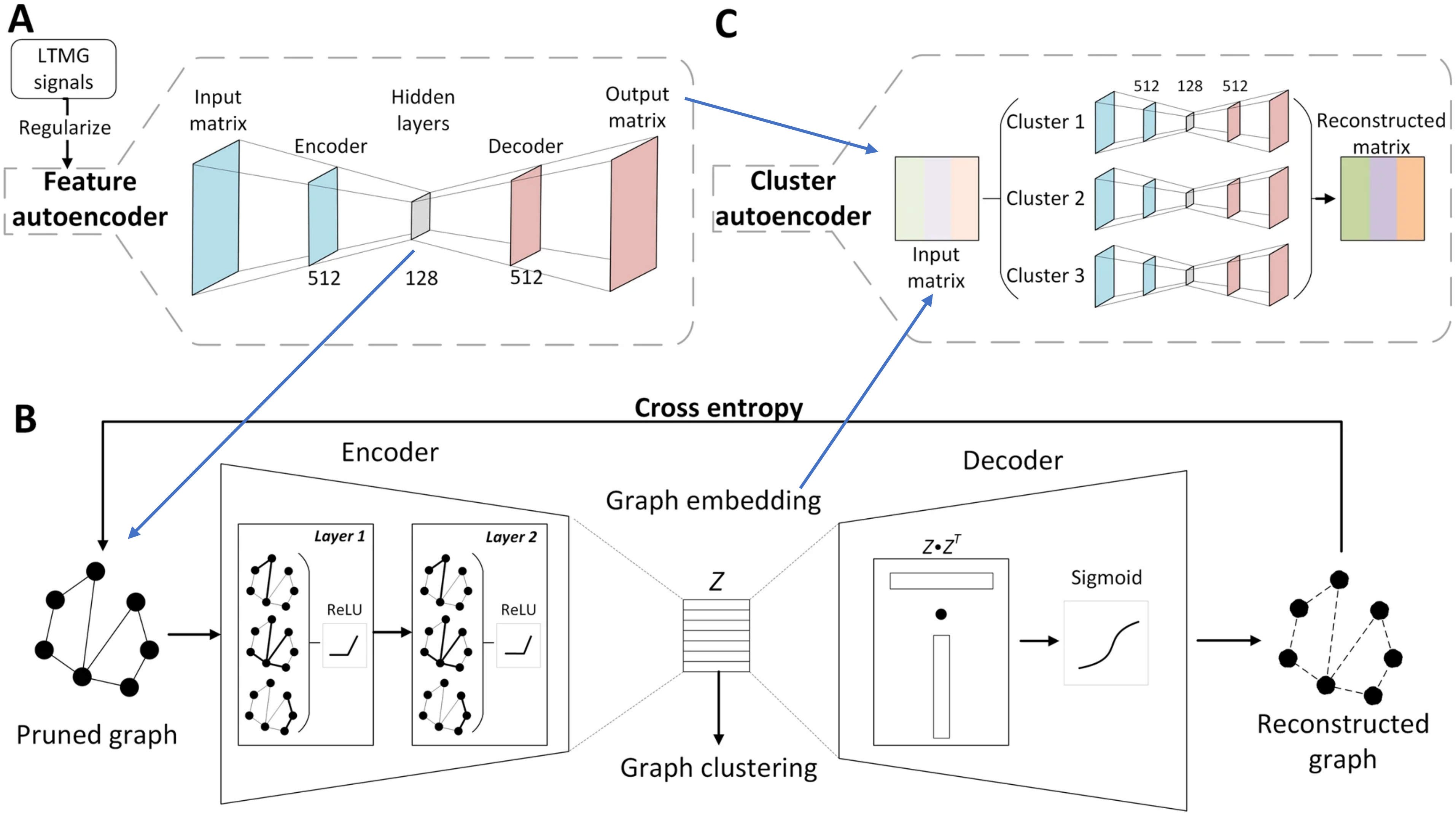
ML in gene expression imputation

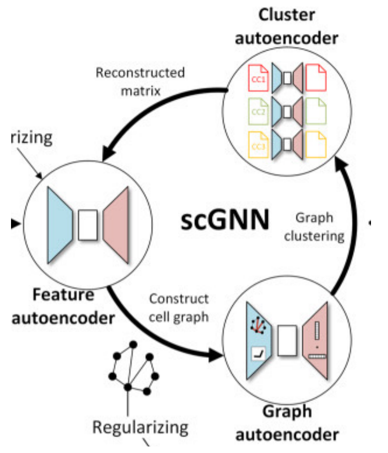
The architecture of scGNN
Wang et al 2021 NC



ML in gene expression imputation

The architecture of scGNN
Wang et al 2021 NC



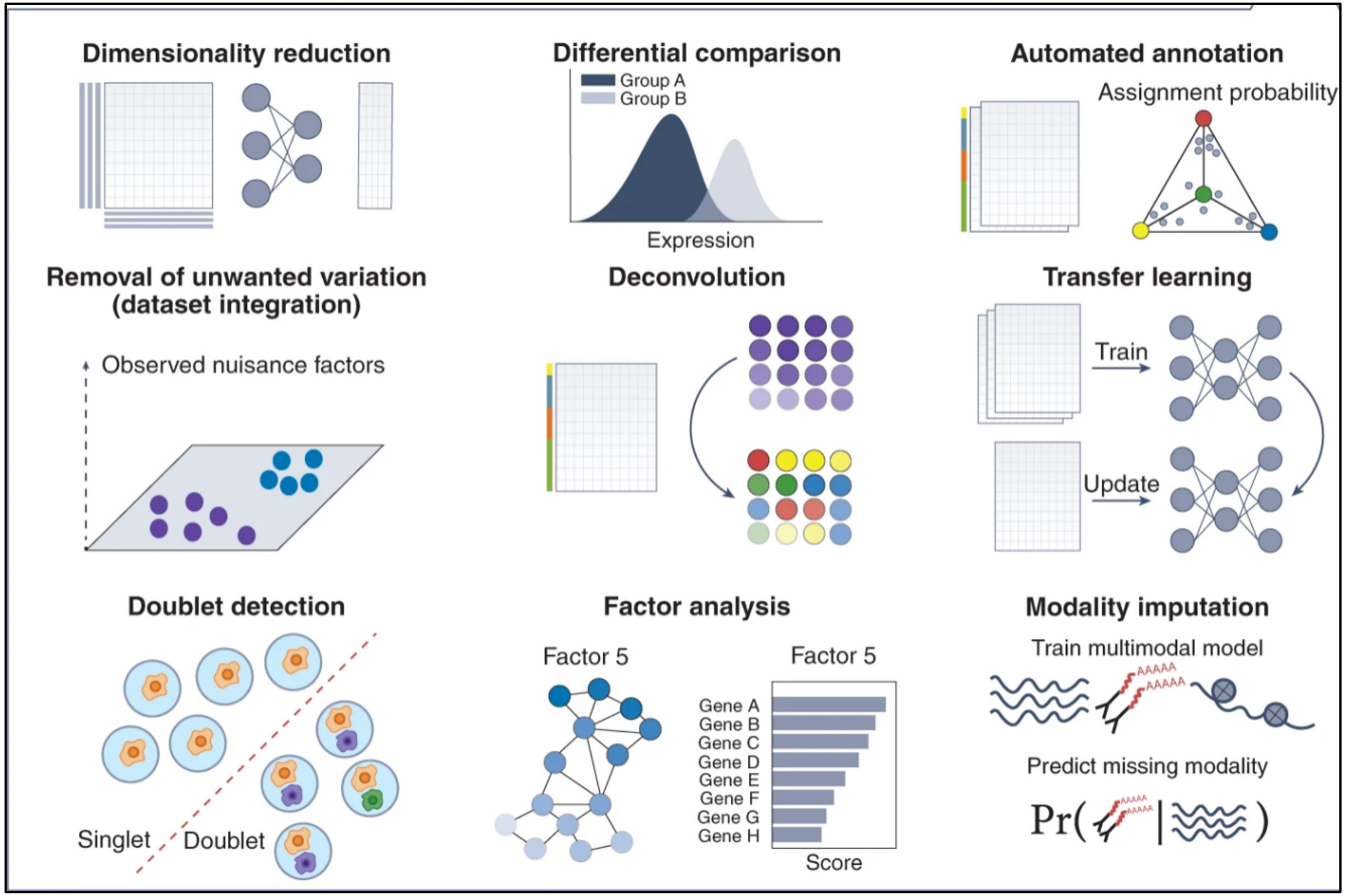


ML in cell classification

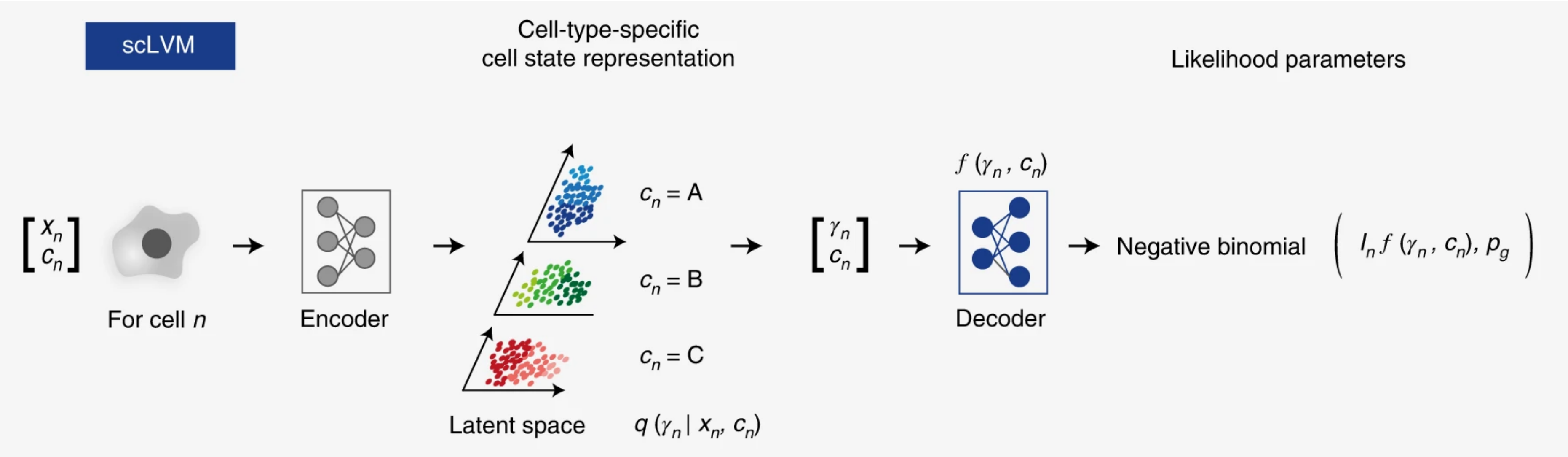
scVI (Lopez et al., 2018)

Tool kit for modelling single-cell-like data using neural networks+probabilistic models

Functions:



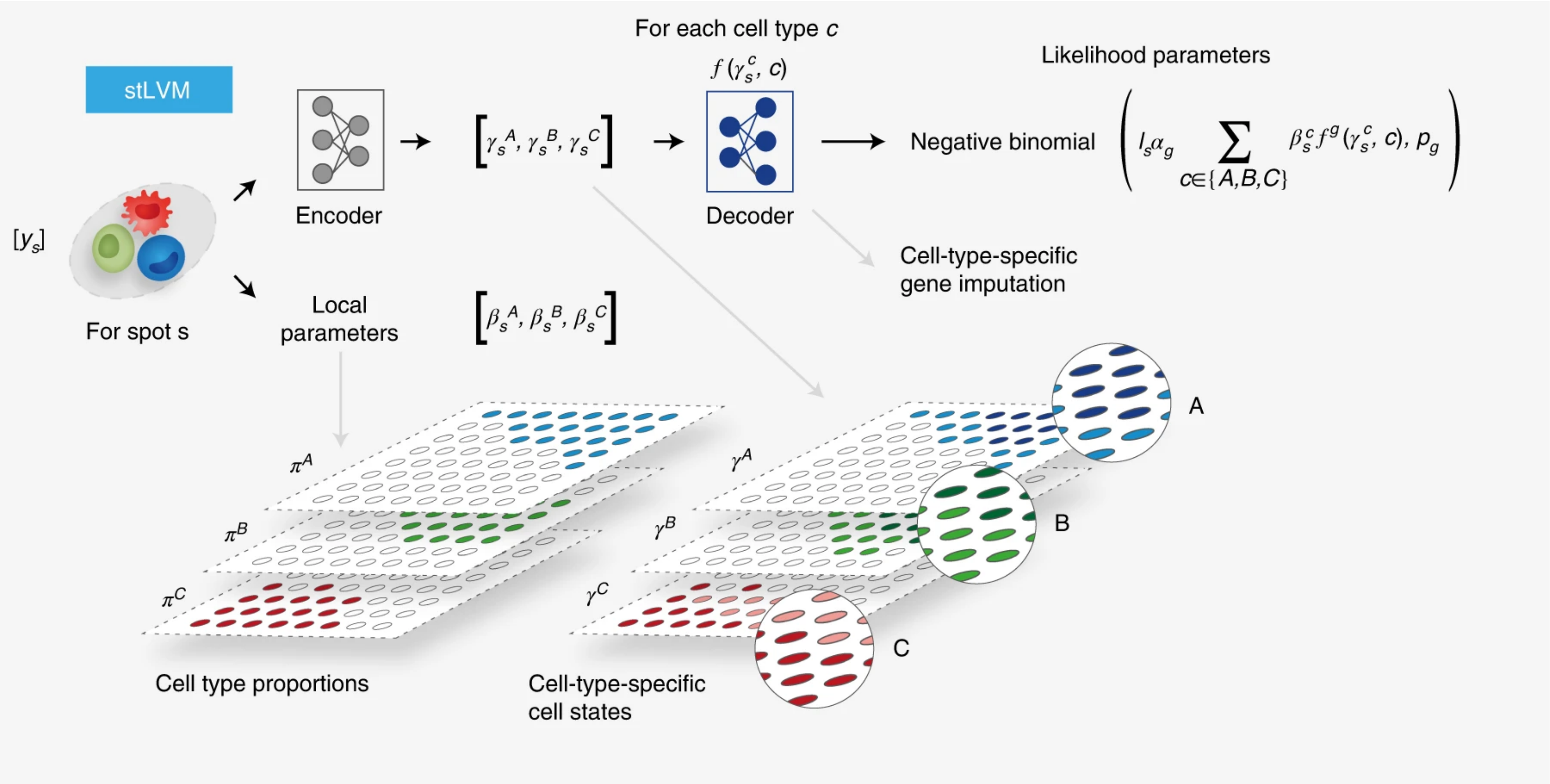
scLVM


scLVM
Cell-type-specific cell state representation
Likelihood parameters
$\begin{bmatrix} x_n \\ c_n \end{bmatrix}$
For cell n
Encoder
c_n = A
c_n = B
c_n = C
Latent space
$q(\gamma_n \mid x_n, c_n)$
$\begin{bmatrix} \gamma_n \\ c_n \end{bmatrix}$
$f(\gamma_n, c_n)$
Decoder
Negative binomial
$\left(l_n f(\gamma_n, c_n), p_g \right)$

$$\gamma_n \sim Normal(0, I)$$

$$x_{ng} \sim NegativeBinomial\left(l_n f^g\left(c_n, \gamma_n\right), p_g\right),$$

stLVM

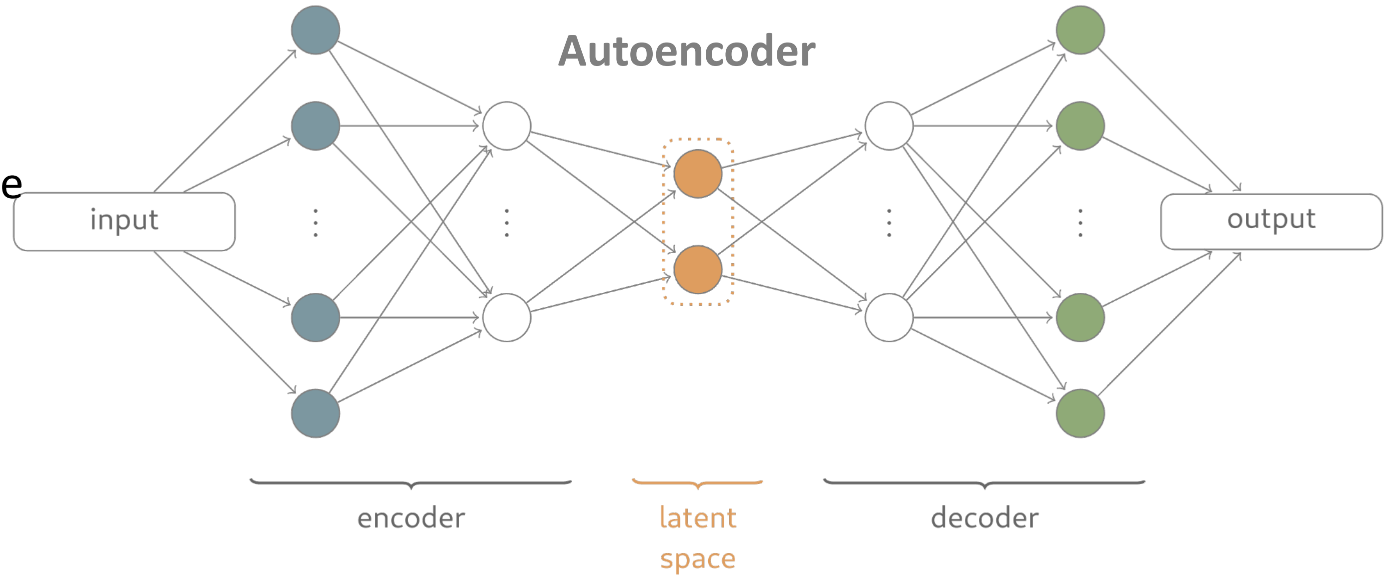

stLVM
$[y_s]$
For spot s
Encoder
Local parameters
$[\gamma_s^A, \gamma_s^B, \gamma_s^C]$
$[\beta_s^A, \beta_s^B, \beta_s^C]$
For each cell type c
$f(\gamma_s^c, c)$
Decoder
Likelihood parameters
Negative binomial $\left(l_s \alpha_g \sum_{c \in \{A,B,C\}} \beta_s^c f^g(\gamma_s^c, c), p_g \right)$
Cell-type-specific gene imputation
π^A
π^B
π^C
Cell type proportions
γ^A
γ^B
γ^C
A
B
C
Cell-type-specific cell states

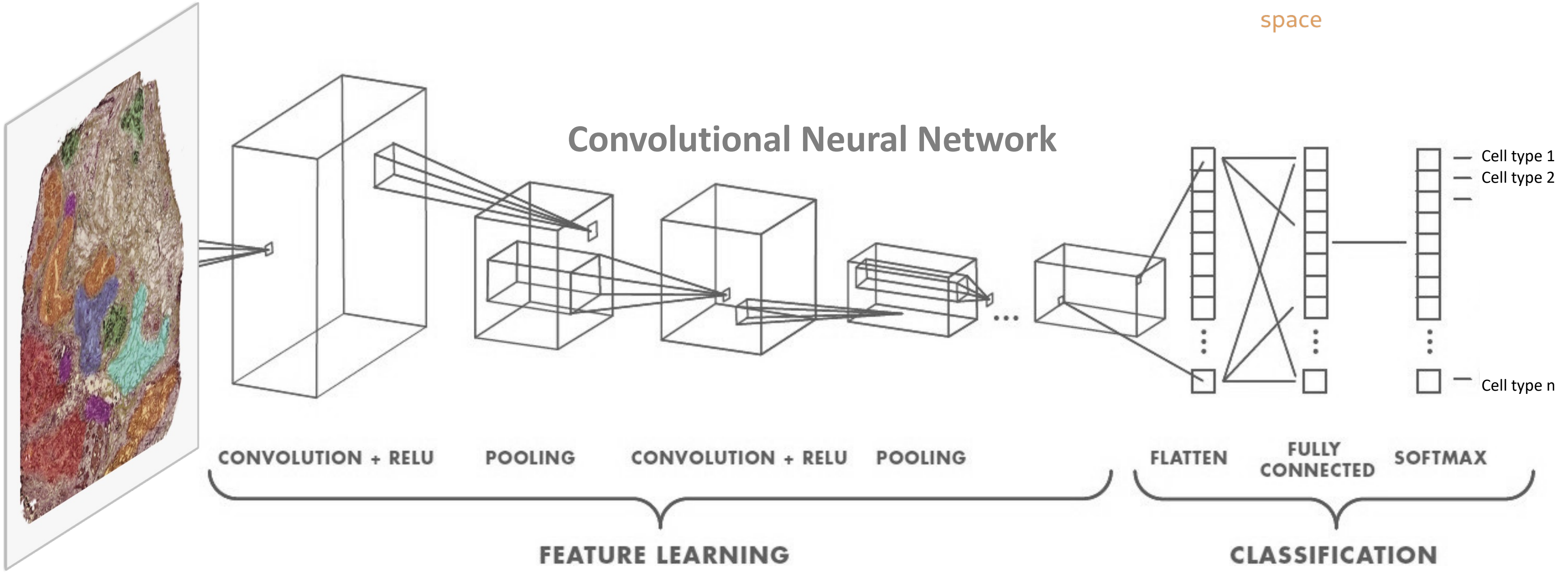
Machine Learning for Spatial Transcriptomics

New analysis: Neural Network for Spatial Transcriptomics

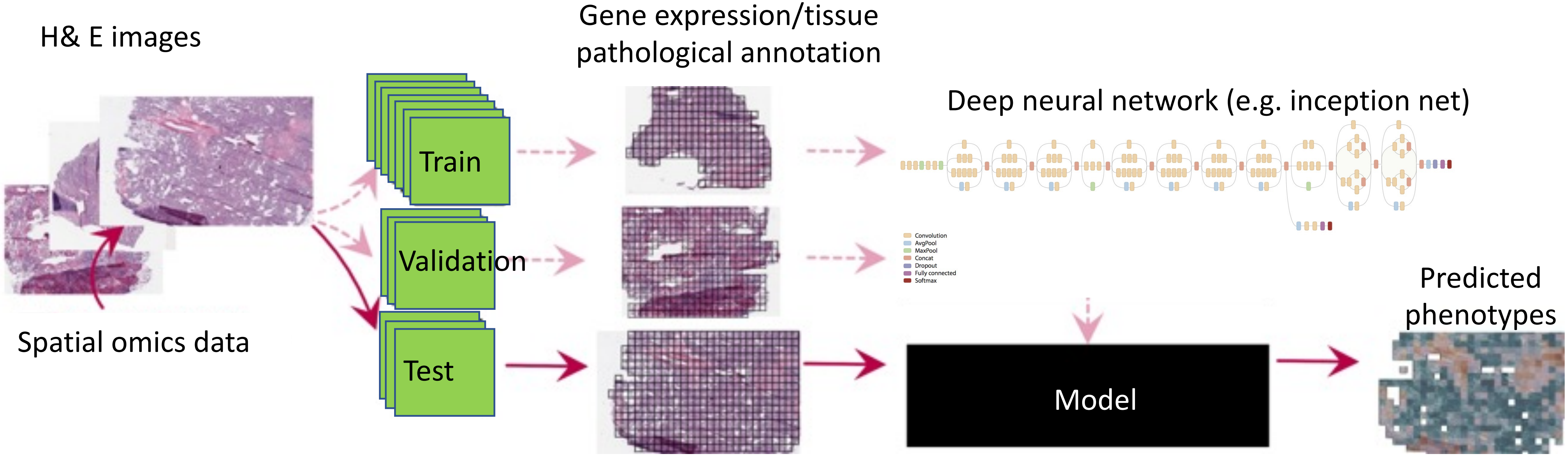
Two neural network (NN) architectures

- Convolutional Neural Network (CNN) for feature extraction
  - Designed for spatial imaging data
- Autoencoder (AE) for combining data
  - Find informative shared latent space





Neural Network Utilizing Molecular Labels



H& E images

Spatial omics data

Train

Validation

Test

Gene expression/tissue pathological annotation

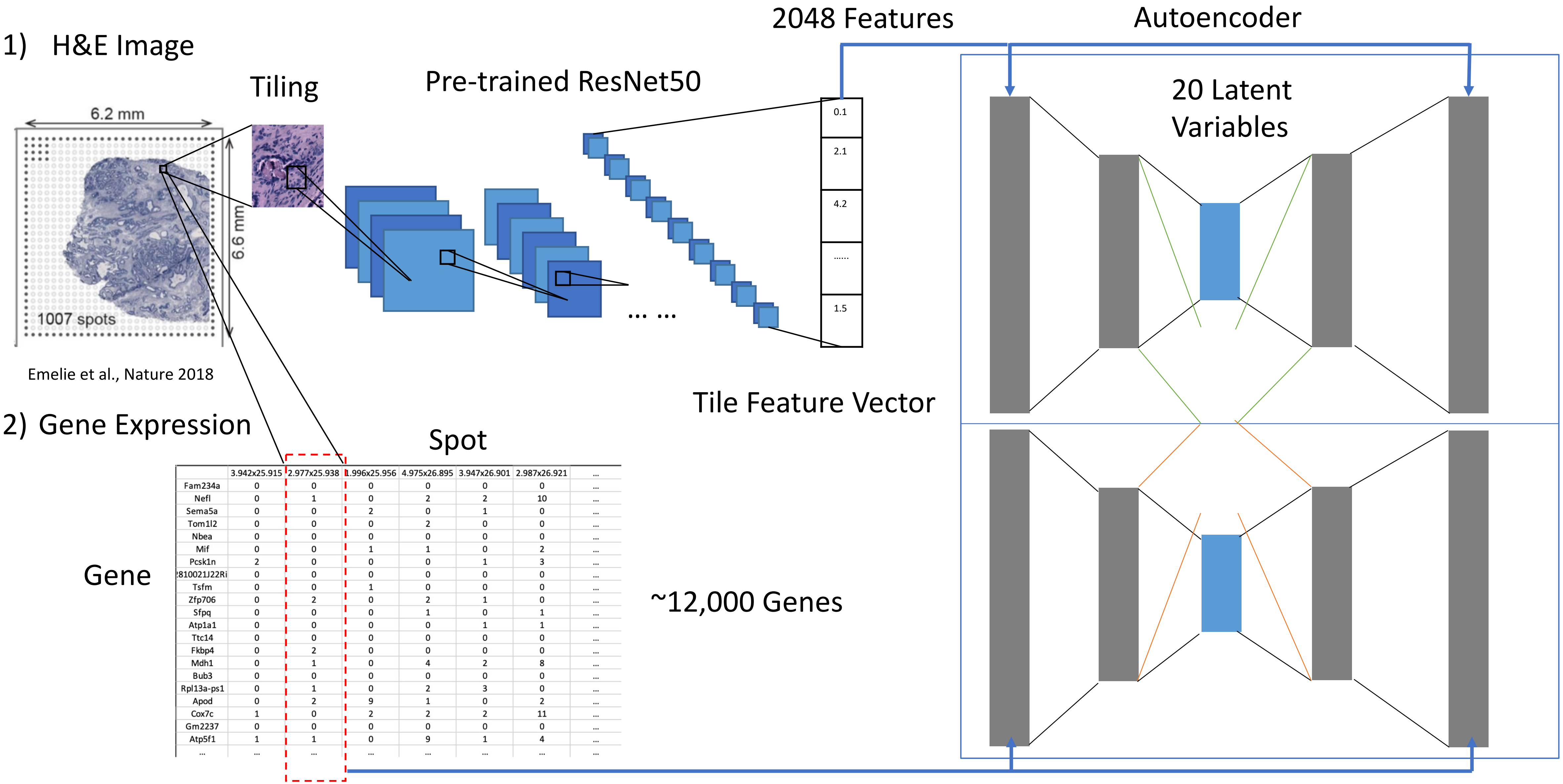
Deep neural network (e.g. inception net)

Model

Predicted phenotypes

- Traditional NN methods using histopathological images rely on tissue-region annotation defined by trained pathologists
- The regional annotation is not accurate at single-cell or pixel levels

Neural networks to analyse spatial transcriptomics data

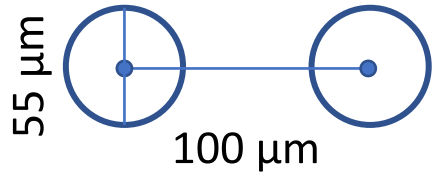


Emelie et al., Nature 2018

	3.942x25.915	2.977x25.938	1.996x25.956	4.975x26.895	3.947x26.901	2.987x26.921	...
Fam234a	0	0	0	0	0	0	...
Nefl	0	1	0	2	2	10	...
Sema5a	0	0	2	0	1	0	...
Tom1l2	0	0	0	2	0	0	...
Nbea	0	0	0	0	0	0	...
Mif	0	0	1	1	0	2	...
Pcsk1n	2	0	0	0	1	3	...
2810021J22Ri	0	0	0	0	0	0	...
Tsfm	0	0	1	0	0	0	...
Zfp706	0	2	0	2	1	0	...
Sfpq	0	0	0	1	0	1	...
Atp1a1	0	0	0	0	1	1	...
Ttc14	0	0	0	0	0	0	...
Fkbp4	0	2	0	0	0	0	...
Mdh1	0	1	0	4	2	8	...
Bub3	0	0	0	0	0	0	...
Rpl13a-ps1	0	1	0	2	3	0	...
Apod	0	2	9	1	0	2	...
Cox7c	1	0	2	2	2	11	...
Gm2237	0	0	0	0	0	0	...
Atp5f1	1	1	0	9	1	4	...
...	...	...	...	...	...	...	...

Spatial transcriptomics allows for the integration of imaging and sequencing data

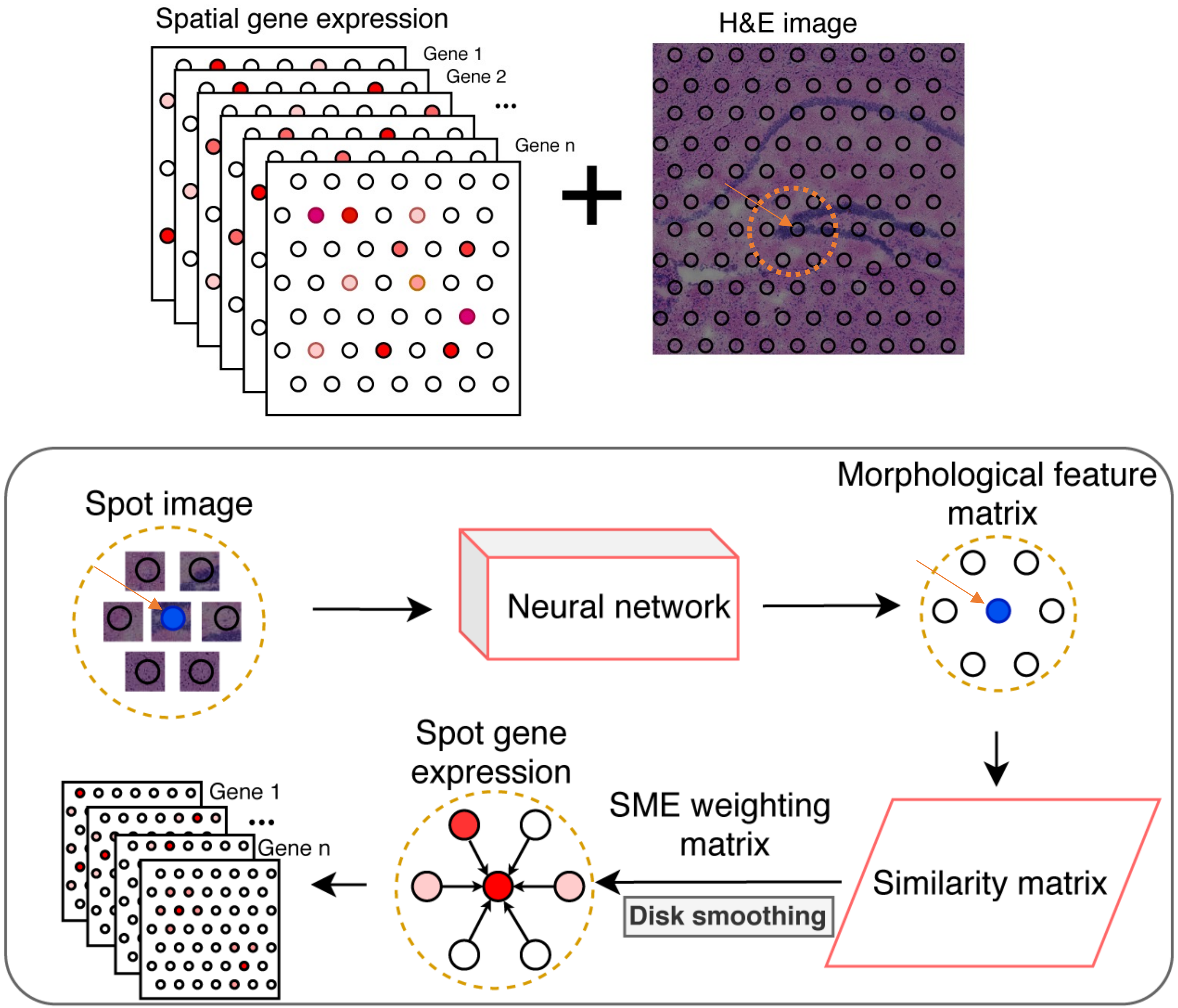
Spatial distance



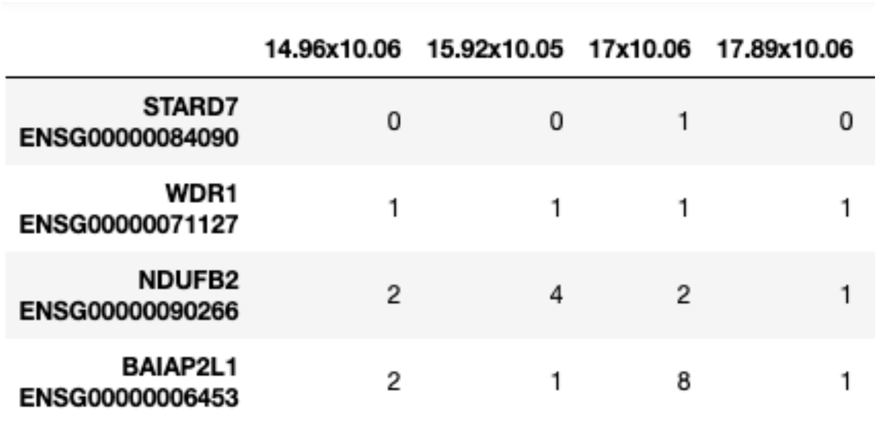
Morphology similariry

Expression values

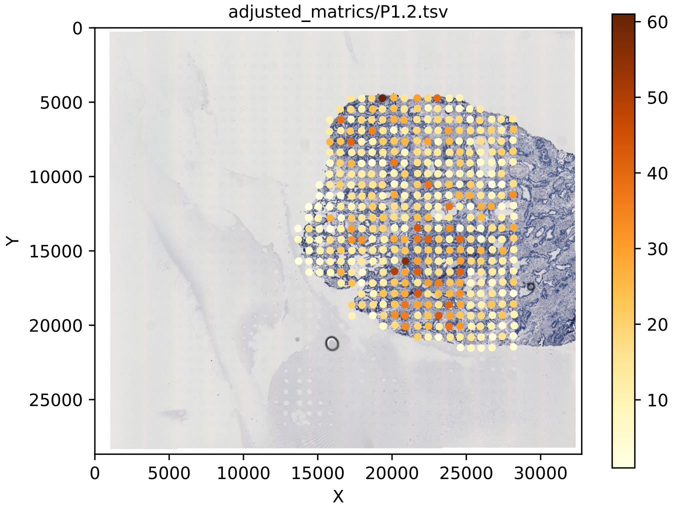
1	**0**
0	5
…	…
3	0
0	0
0	1



Spatial Transcriptomics Data (Slide-seq): expression + location

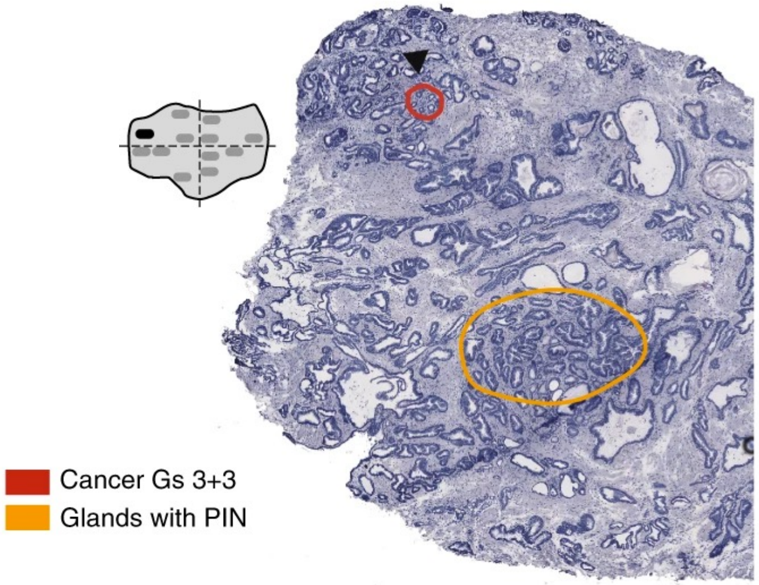


	14.96x10.06	15.92x10.05	17x10.06	17.89x10.06
STARD7 ENSG00000084090	0	0	1	0
WDR1 ENSG00000071127	1	1	1	1
NDUFB2 ENSG00000090266	2	4	2	1
BAIAP2L1 ENSG00000006453	2	1	8	1



Imaging pixel intensity is **NOT** used:

```
for i in range(img.size[0]):
   for j in range(img.size[1]):
      r, g, b = pixels[i,j]
```



(Berglund et al, 2018)

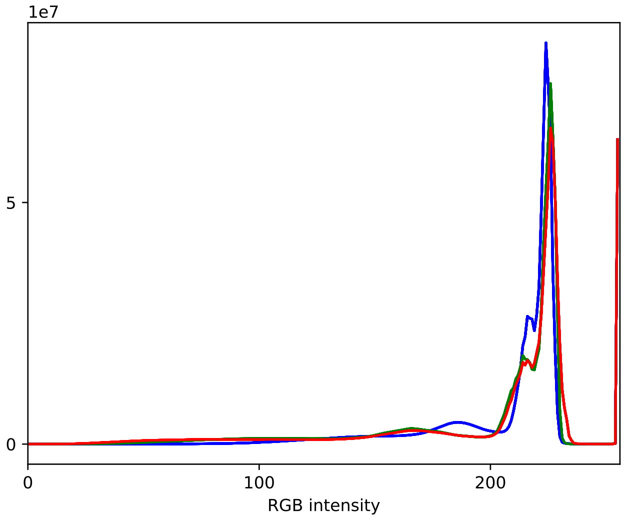


Image mode=RGB, size=32768x28672, (28672, 32768, 3)

Existing analysis methods

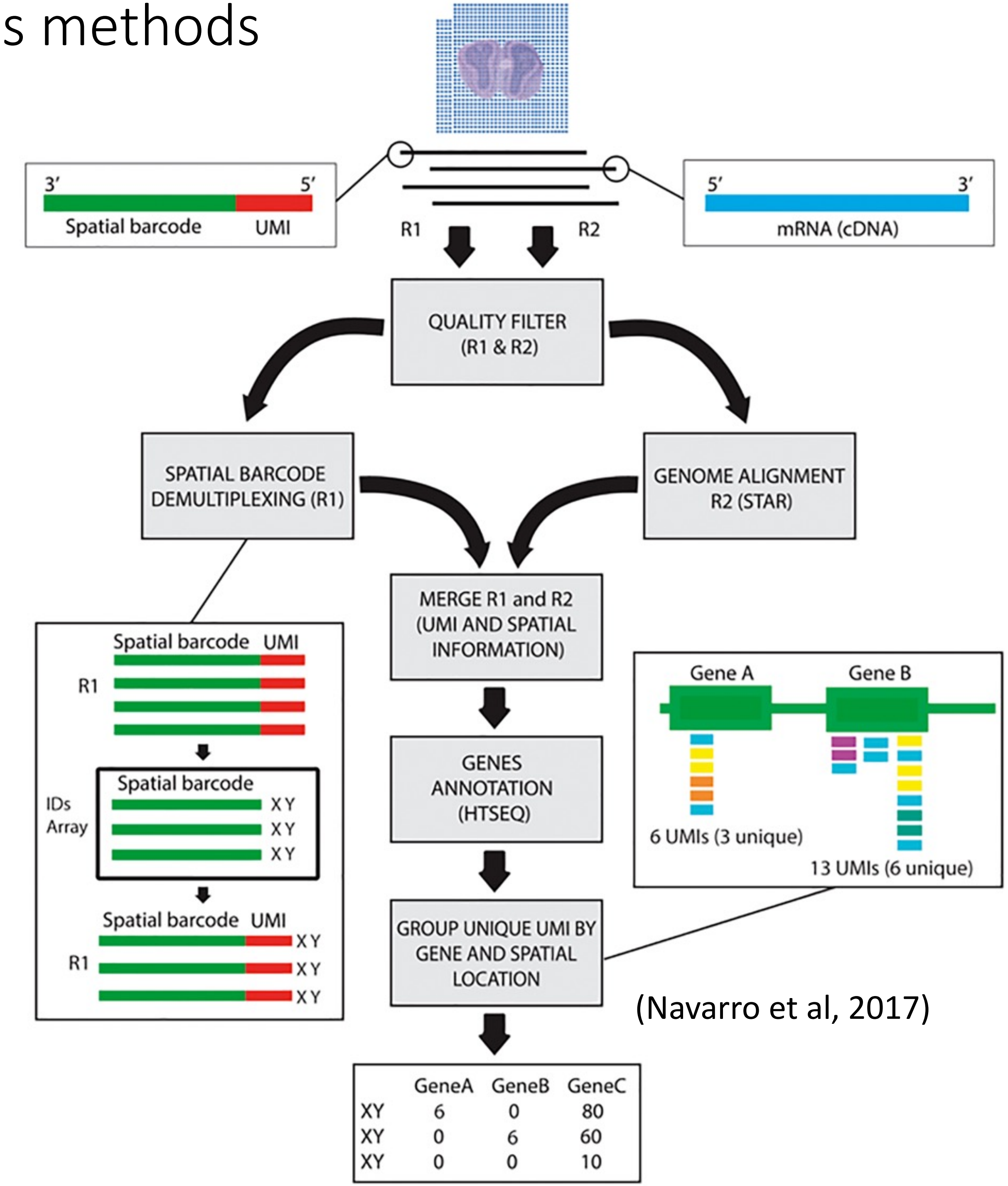
- Preprocessing: genes excluded if not in 10 cells and cells excluded if not having above 10 genes detected

- Normalisation: TMM or RLE method (as in EdgeR), deconvolution by pooling (as in scran), library sizes followed by log transformation, size factors as in DESeq, regress out covariates

- Feature (gene) selection: e.g. highly variable genes

- Dimensionality reduction: PCA followed by UMAP and tSNE

- Clustering and differential expression analysis: similar to single cell data



3′ 5′ Spatial barcode UMI

R1 R2

5′ 3′ mRNA (cDNA)

QUALITY FILTER (R1 & R2)

SPATIAL BARCODE DEMULTIPLEXING (R1)

GENOME ALIGNMENT R2 (STAR)

MERGE R1 and R2 (UMI AND SPATIAL INFORMATION)

GENES ANNOTATION (HTSEQ)

GROUP UNIQUE UMI BY GENE AND SPATIAL LOCATION

Spatial barcode UMI

R1

IDs Array

Spatial barcode XY XY XY

Spatial barcode UMI

R1 XY XY XY

Gene A Gene B

6 UMIs (3 unique)

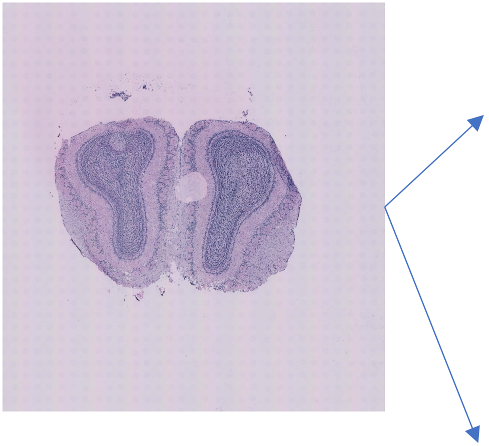
13 UMIs (6 unique)

	GeneA	GeneB	GeneC
XY	6	0	80
XY	0	6	60
XY	0	0	10

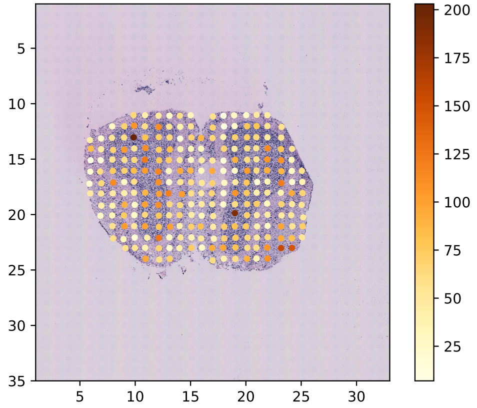
(Navarro et al, 2017)

Existing analysis methods

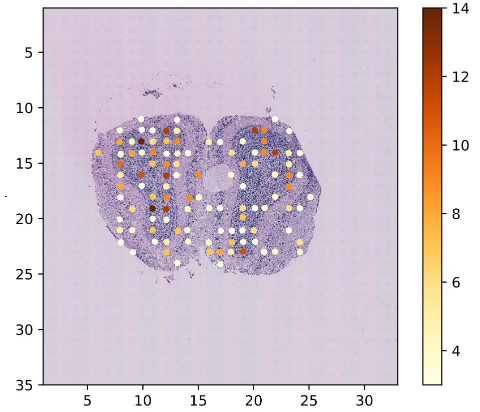
H&E image



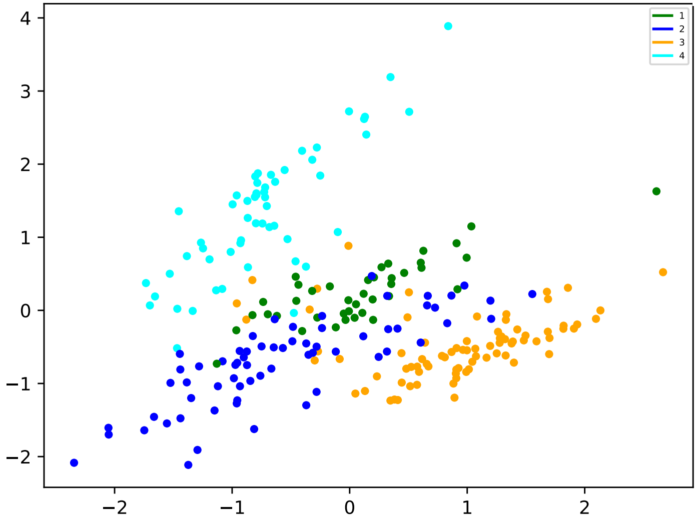
Actb



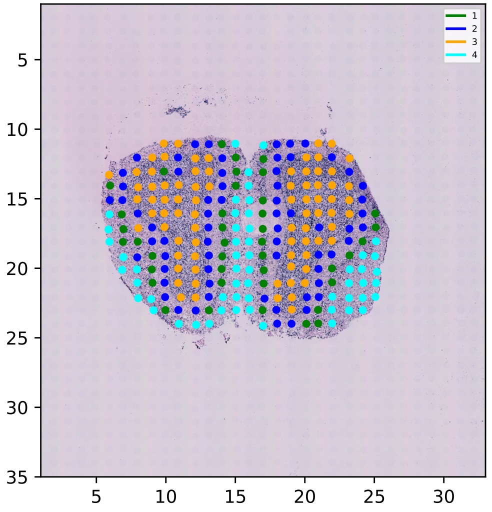
Hoxd8



Clustering cell-spots



Clustering cell-spots on tissue

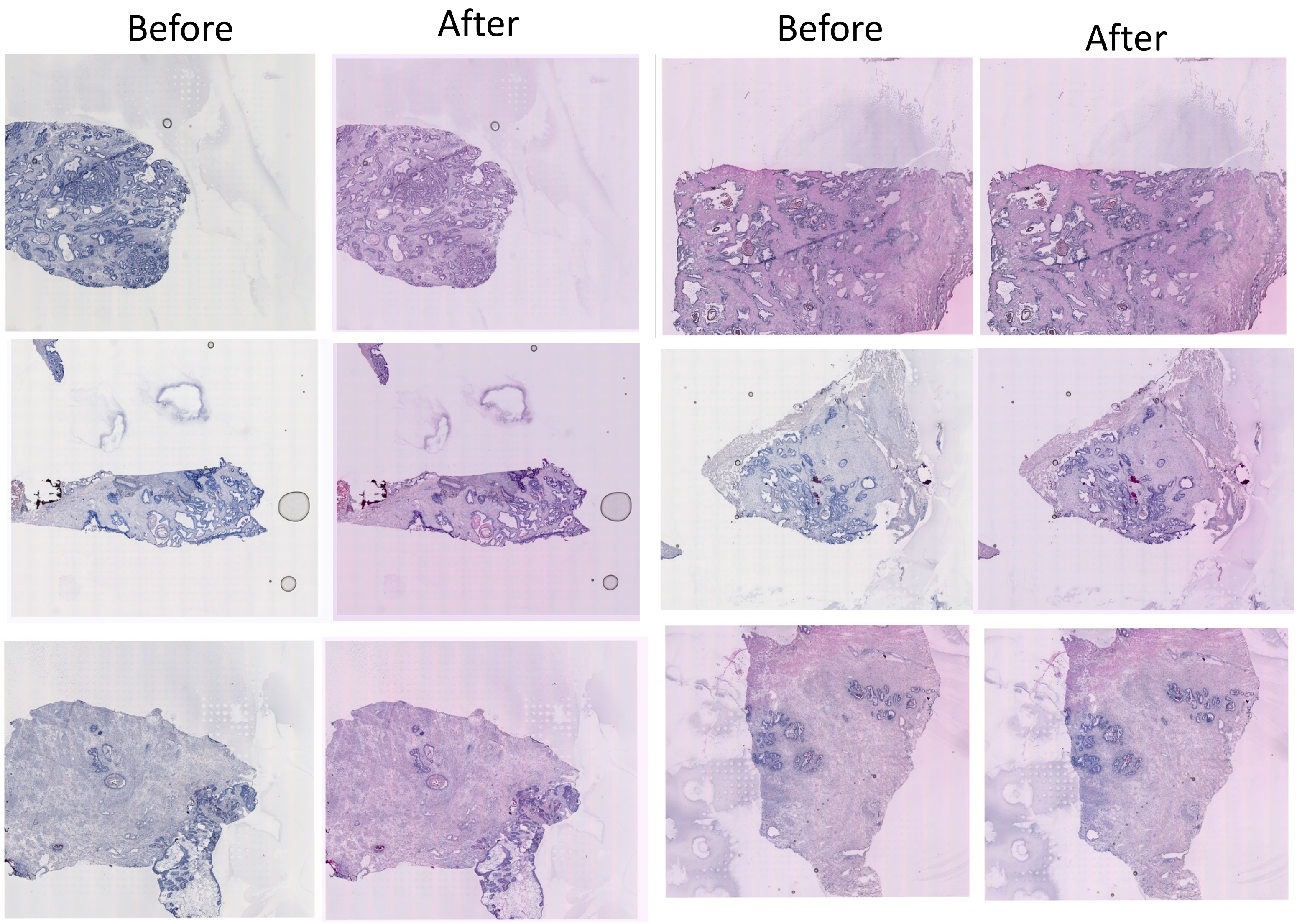


Example of data preprocessing:

- Total number of spots 242
- Total number of genes 16,251
- Dropped 3 spots (too few genes)
- Dropped 1233 genes (detected in too few spots)

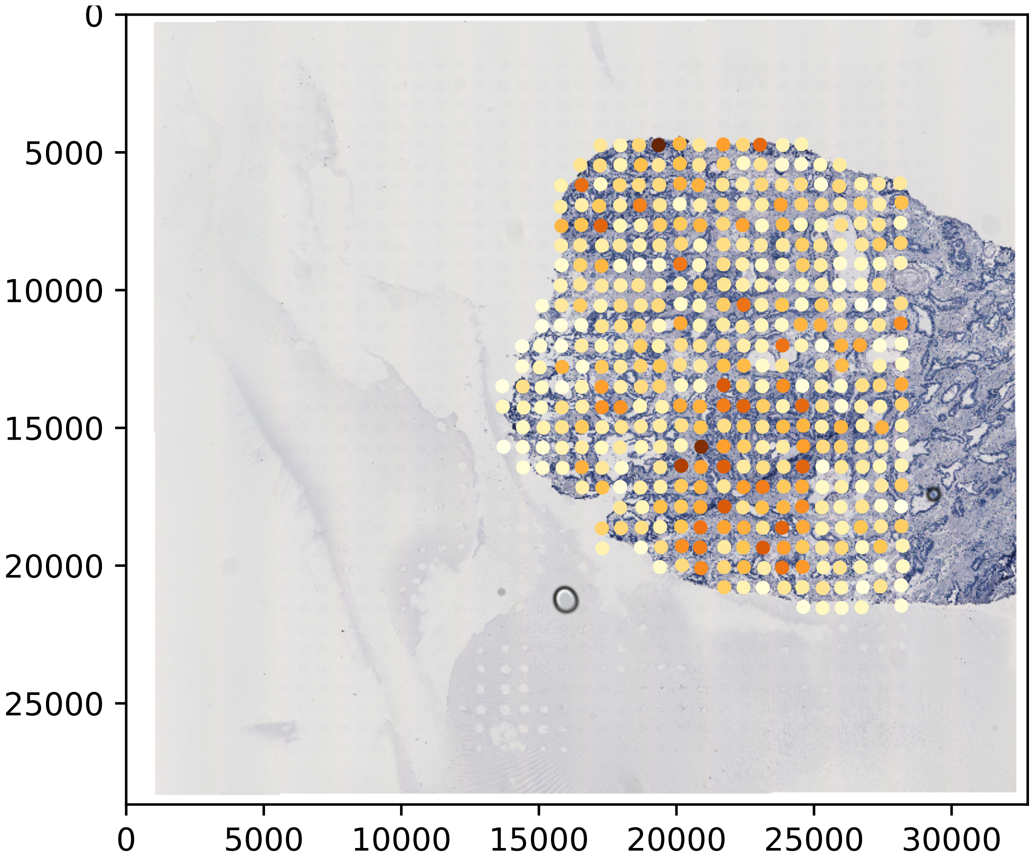
New analysis: Normalisation between images

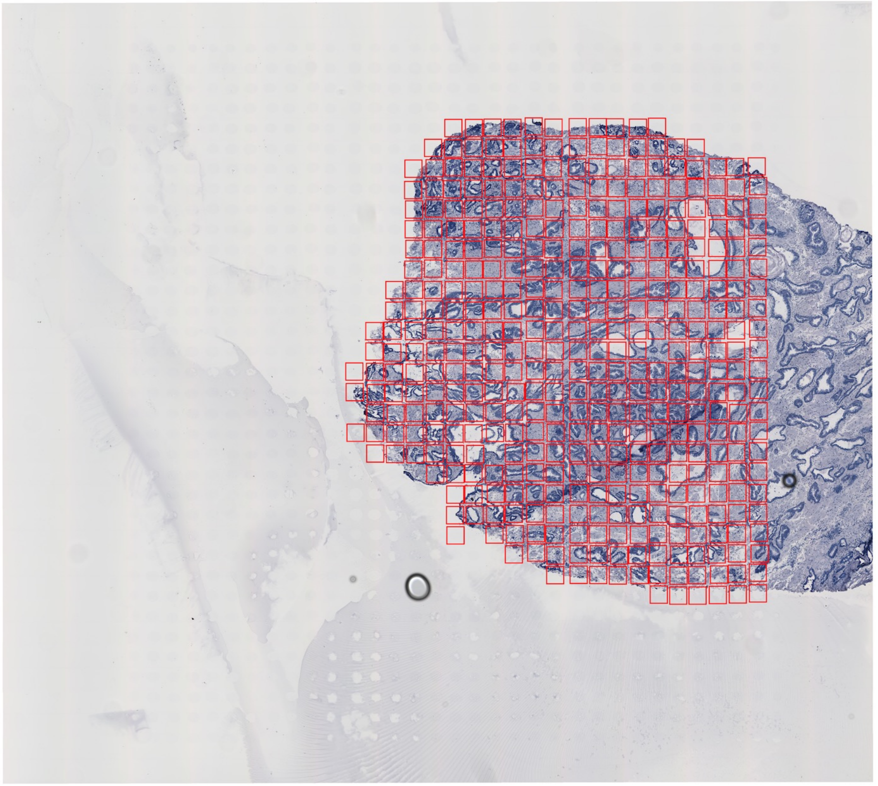
	H&E image
Preprocessing	● Remove low quality images (tissue artifacts) ● Tiling ● Random rotation of tiles: to increase model generalizability
Normalization	● Color cast removal ● Vahadane stain normalization ● Standardization



New analysis: Tiling images to increase sample size

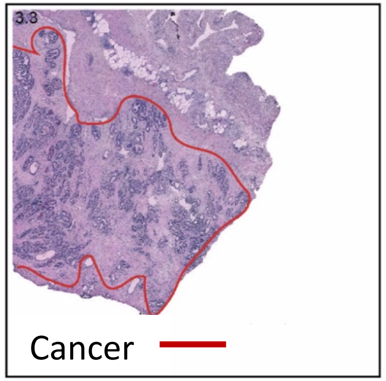
- Each Slide-seq spot corresponds to one tile, which contains both gene expression and H&E image pixel data
- Size of a spot is 299x299 pixels, and thus is represented by a (299, 299, 3) array
- From 12 images, generate 5910 tiles for training data



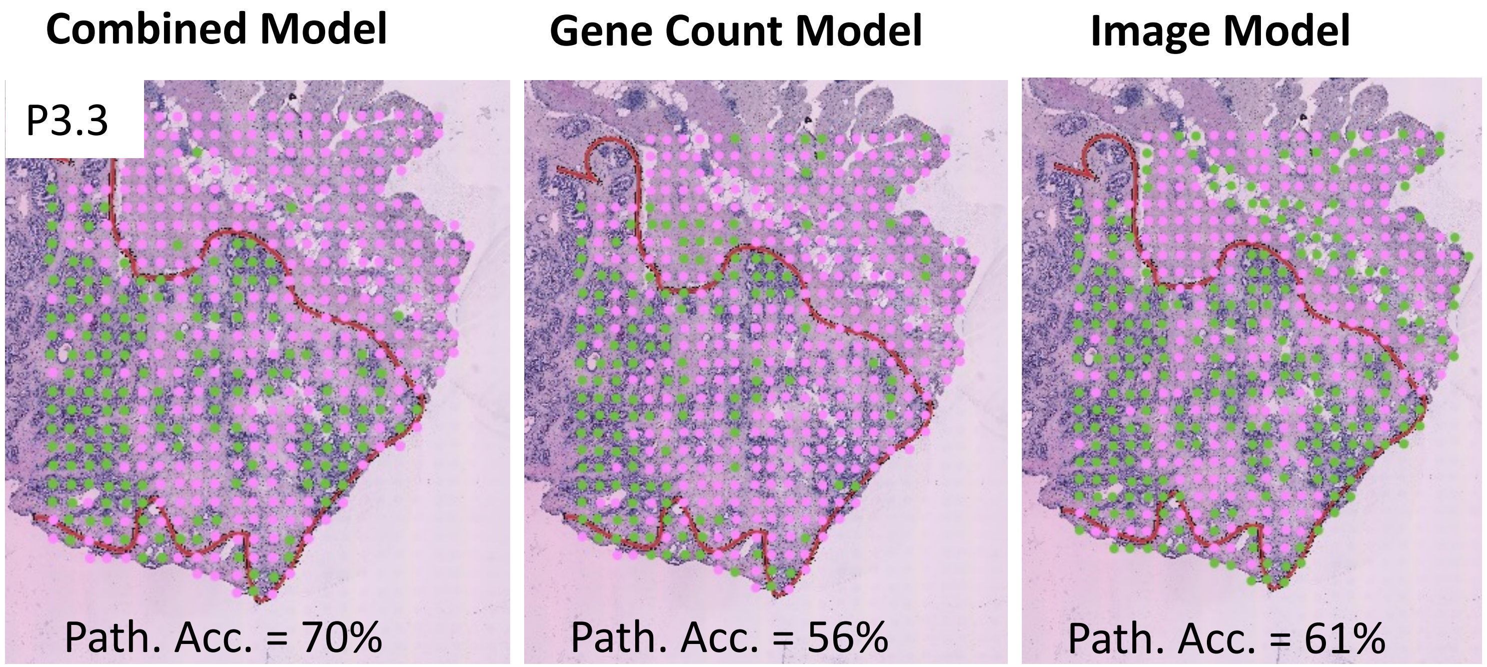


Finding cancer cells by integrating count matrix and imaging data

Pathological Annotation

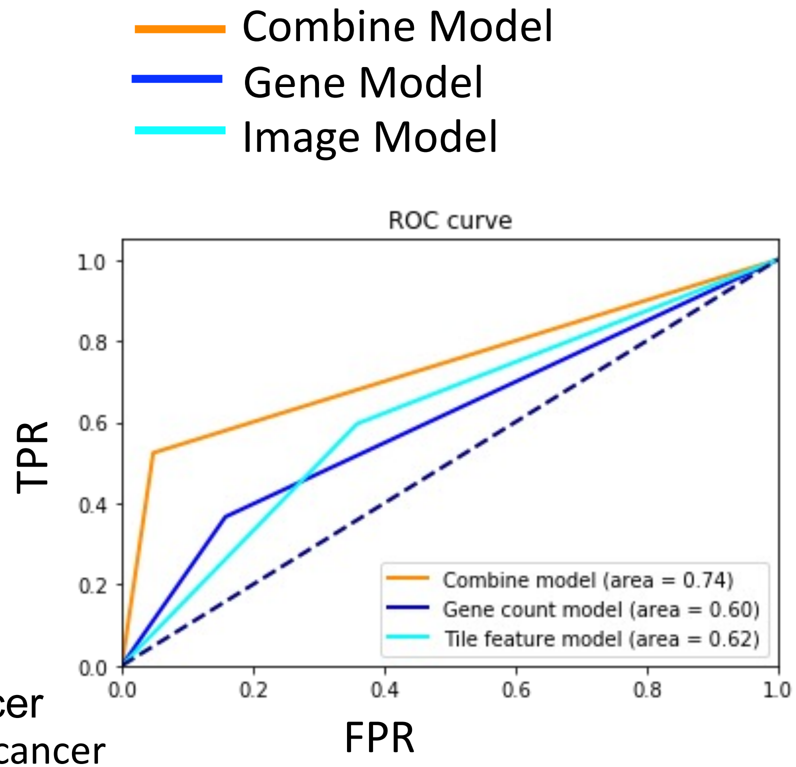


- Combining gene expression and image information is better than using gene expression or image alone
- Typical pathological annotation by drawing regions on images is not as accurate as computational annotation at pixel level



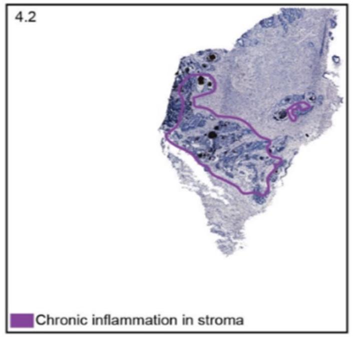
● Cancer

● Non-cancer



Finding inflamed stromal cells by integrating count matrix and imaging data

Pathological Annotation



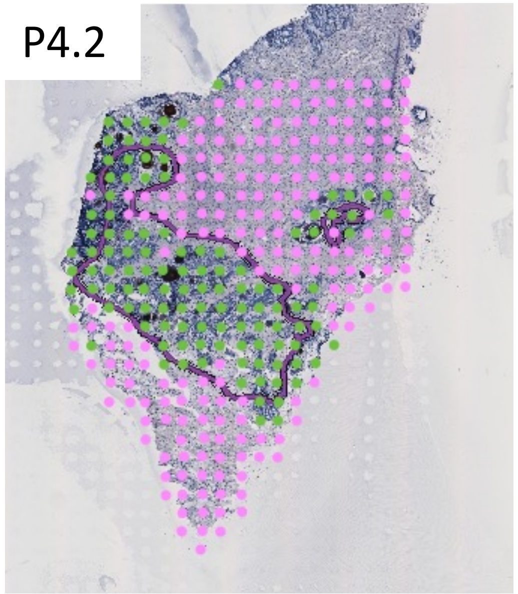
Inflamed stromal cells

- The Tissue image + Gene count combination resulted in lower false positive spots
- Sensitive to detect a small inflamed stromal cell region

● Inflamed stromal
● Normal

— Combine Model
— Gene Model
— Image Model

Combined Model



Gene Count Model

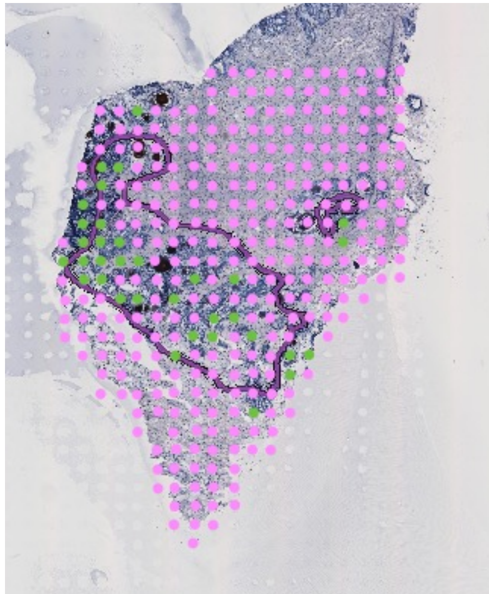
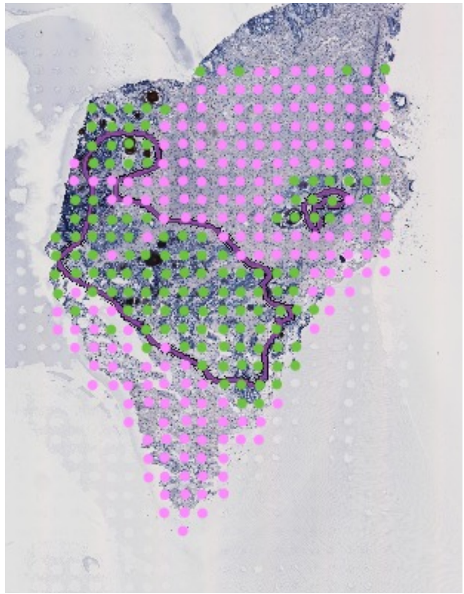
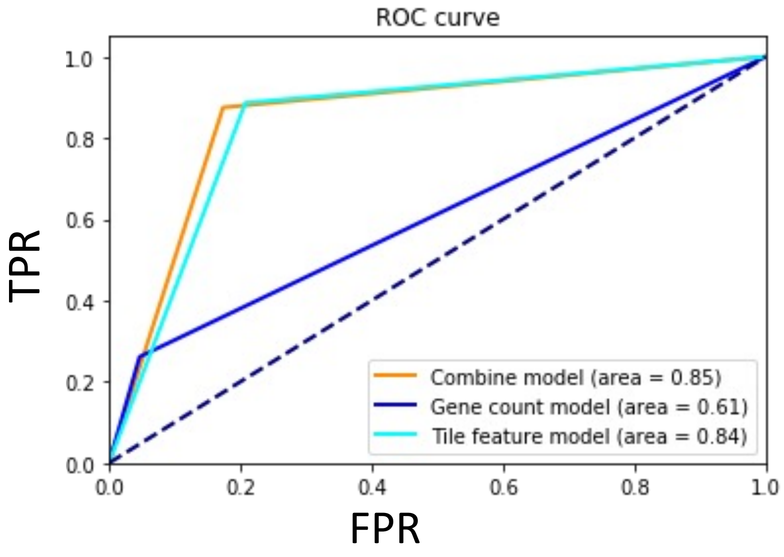


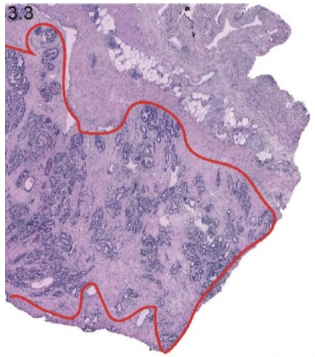
Image Model



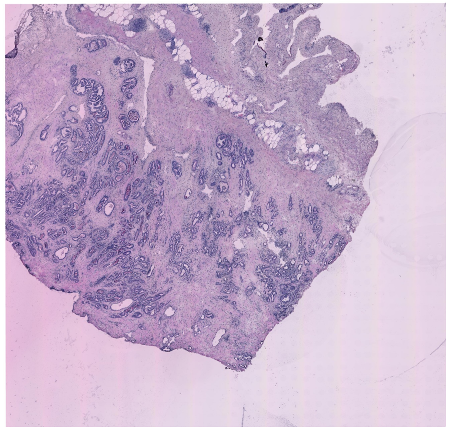

ROC curve
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Combine model (area = 0.85)
Gene count model (area = 0.61)
Tile feature model (area = 0.84)

Quantitative Validation

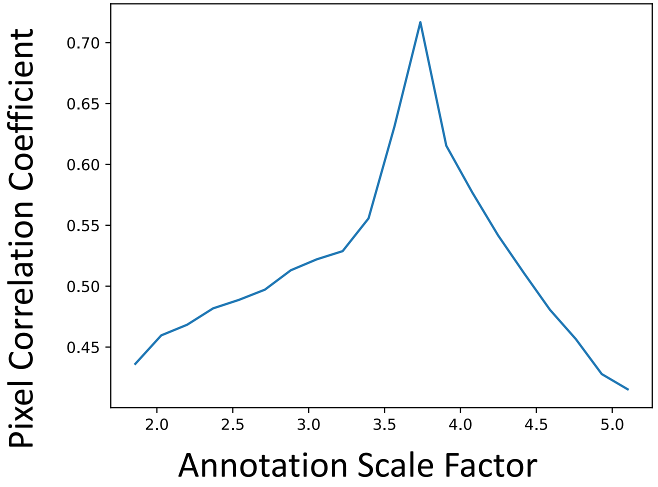
Pathological Annotation (PA)



Whole Slide Image (WSI)

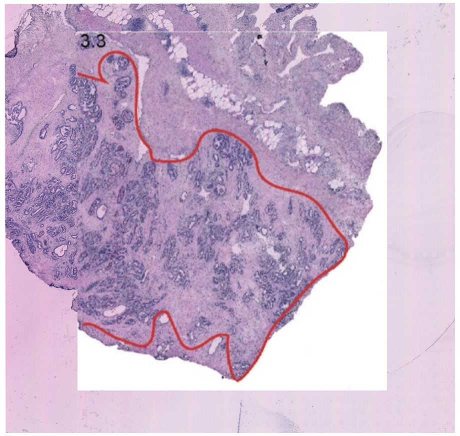


Registration

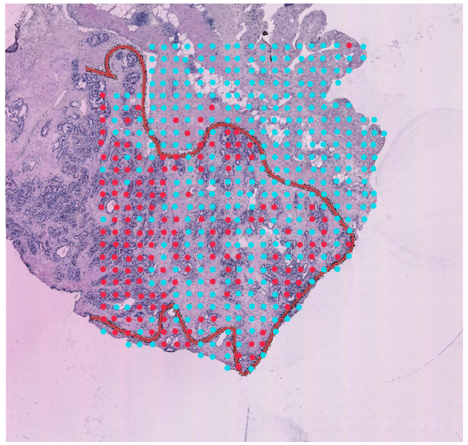


PA mapped on WSI

Contour mapped on WSI

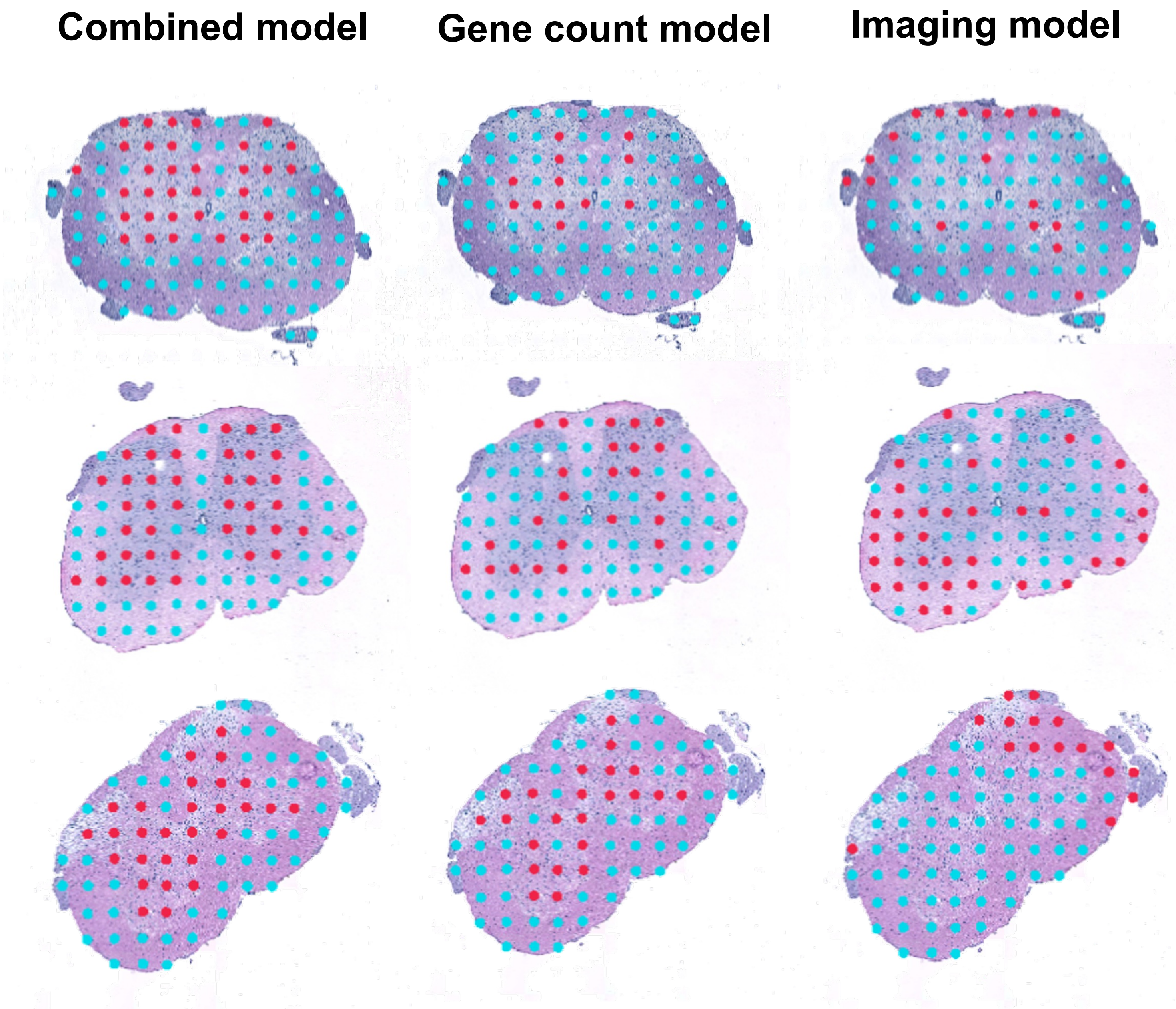


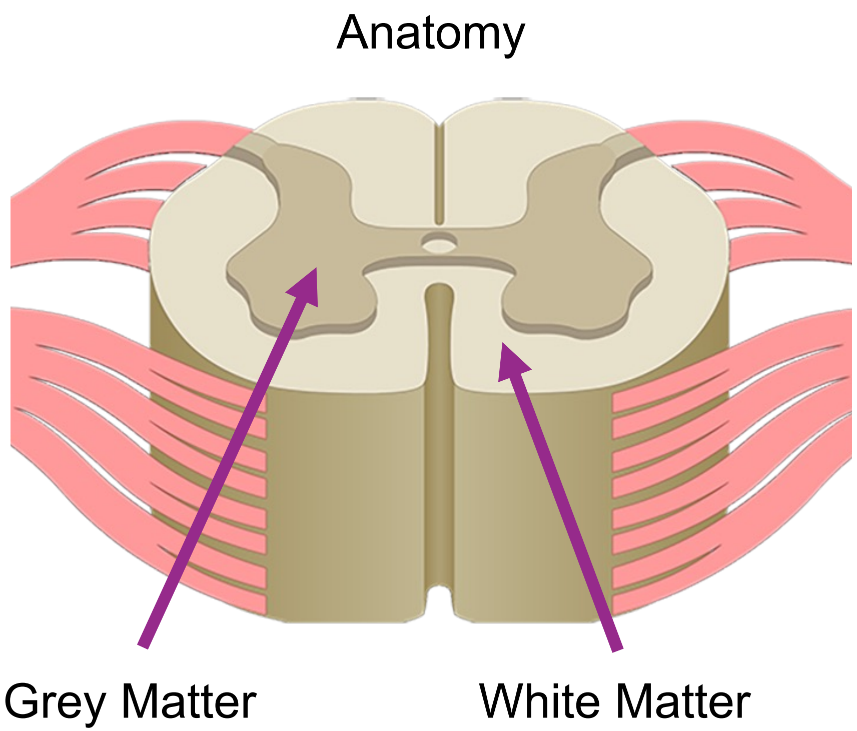
Spot cluster + contour mapped on WSI



+ Quantitative Performance Metrics

Classification of Anatomical Spatial Regions





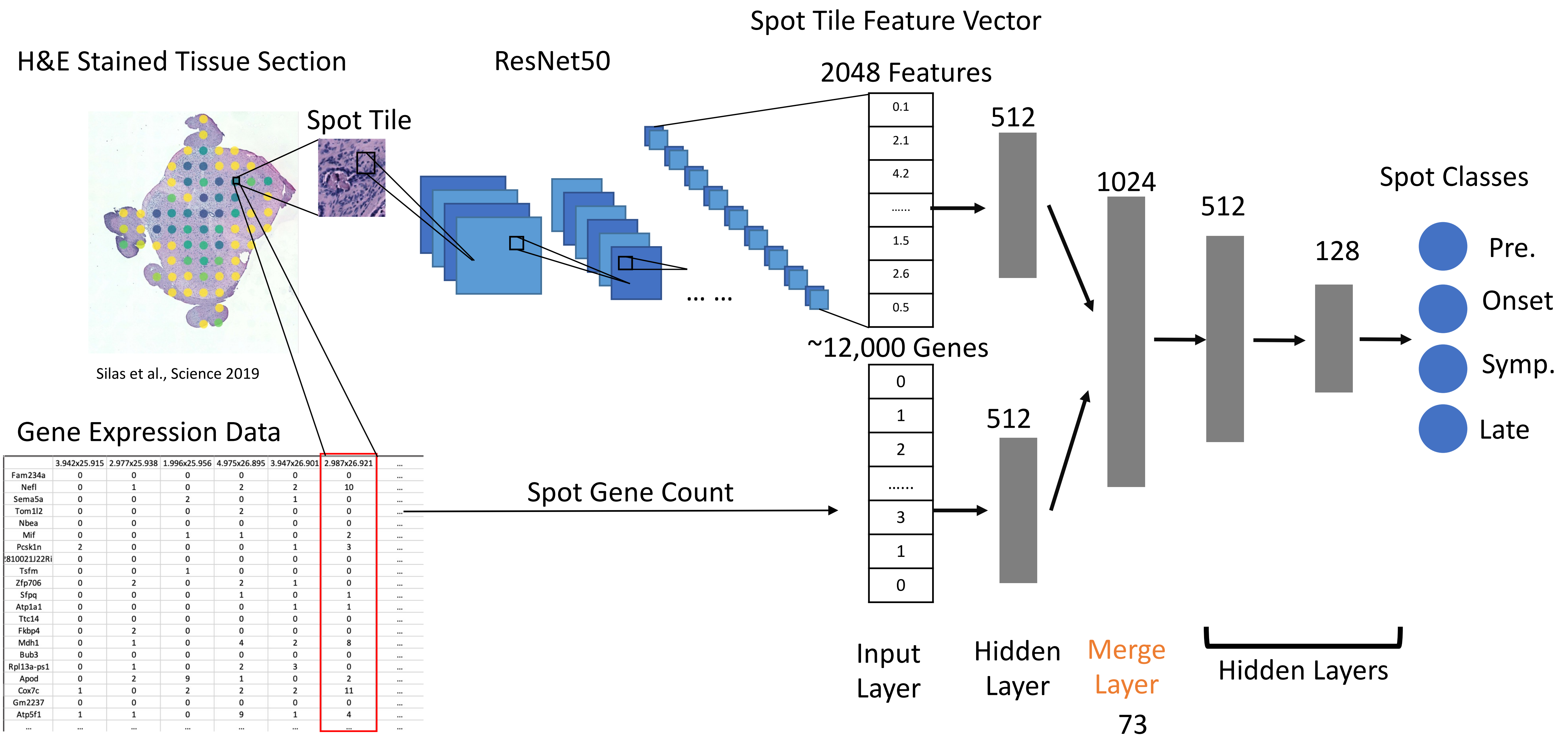
Combining count data and imaging data increases the accuracy of grey and white matter classification

● Cluster 1

● Cluster 2

Silas et al., Science 2019

Disease Stage Classification Model



Silas et al., Science 2019

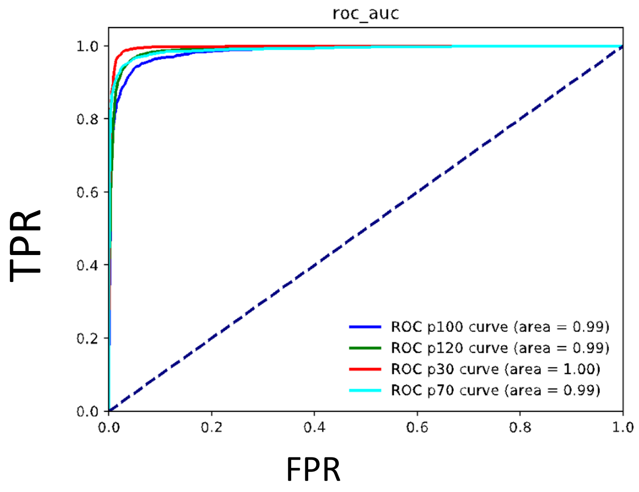
Gene Expression Data

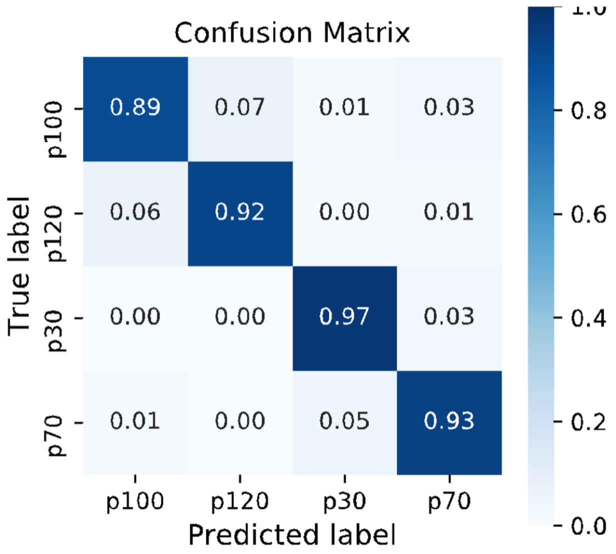
	3.942x25.915	2.977x25.938	1.996x25.956	4.975x26.895	3.947x26.901	2.987x26.921	...
Fam234a	0	0	0	0	0	0	...
Nefl	0	1	0	2	2	10	...
Sema5a	0	0	2	0	1	0	...
Tom1l2	0	0	0	2	0	0	...
Nbea	0	0	0	0	0	0	...
Mif	0	0	1	1	0	2	...
Pcsk1n	2	0	0	0	1	3	...
2810021J22Ri	0	0	0	0	0	0	...
Tsfm	0	0	1	0	0	0	...
Zfp706	0	2	0	2	1	0	...
Sfpq	0	0	0	1	0	1	...
Atp1a1	0	0	0	0	1	1	...
Ttc14	0	0	0	0	0	0	...
Fkbp4	0	2	0	0	0	0	...
Mdh1	0	1	0	4	2	8	...
Bub3	0	0	0	0	0	0	...
Rpl13a-ps1	0	1	0	2	3	0	...
Apod	0	2	9	1	0	2	...
Cox7c	1	0	2	2	2	11	...
Gm2237	0	0	0	0	0	0	...
Atp5f1	1	1	0	9	1	4	...
...	...	...	...	...	...	...	...

Disease Stage Classification - Performance

Combined model

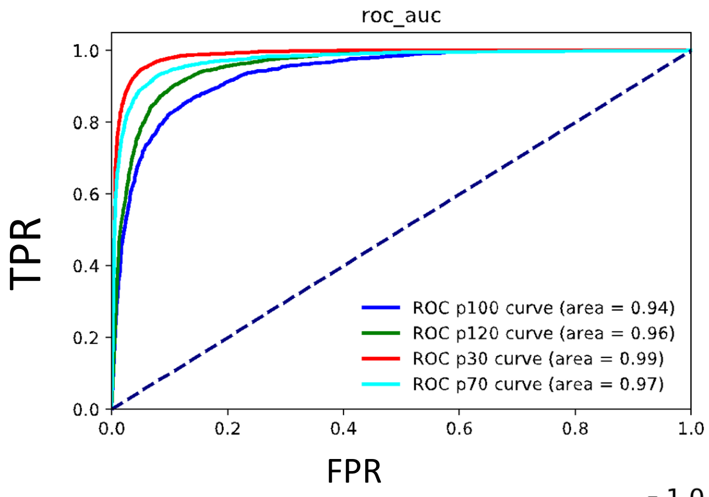
Test accuracy : ~92.75%





Gene count model

Test accuracy: ~84%



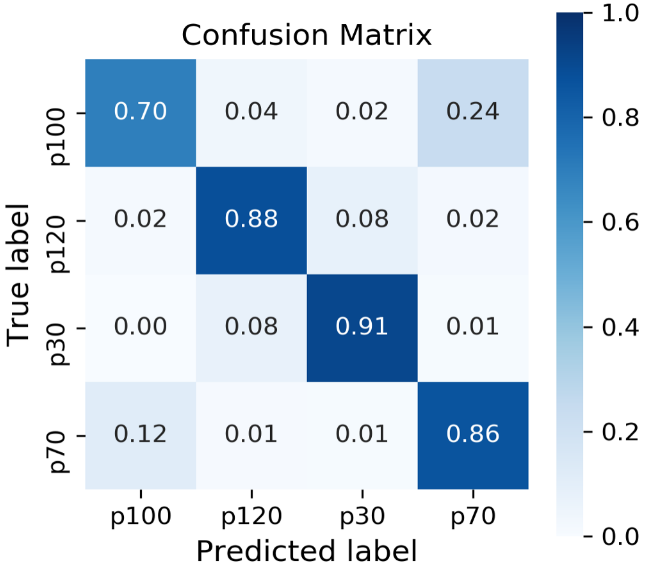
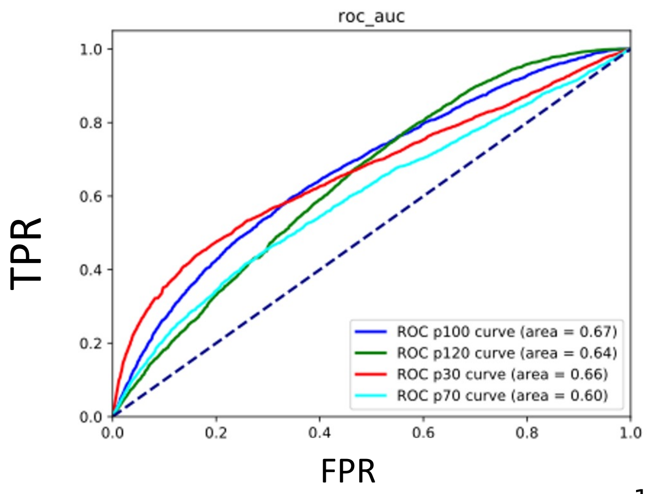
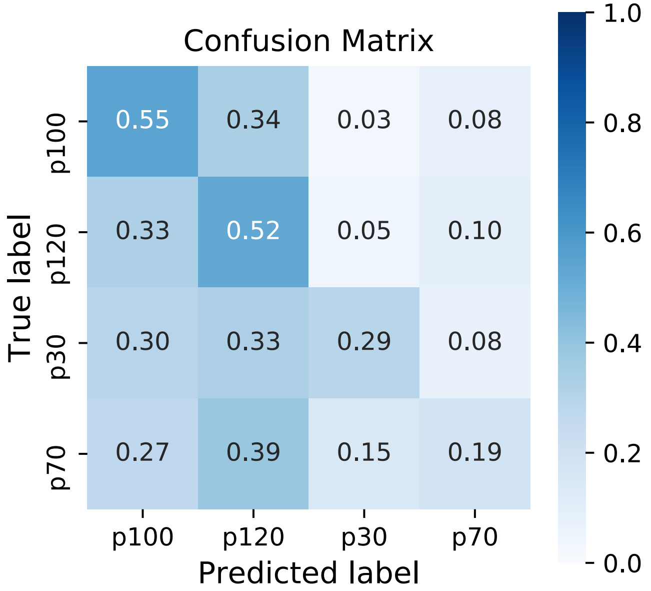


Image model

Test accuracy: ~40%

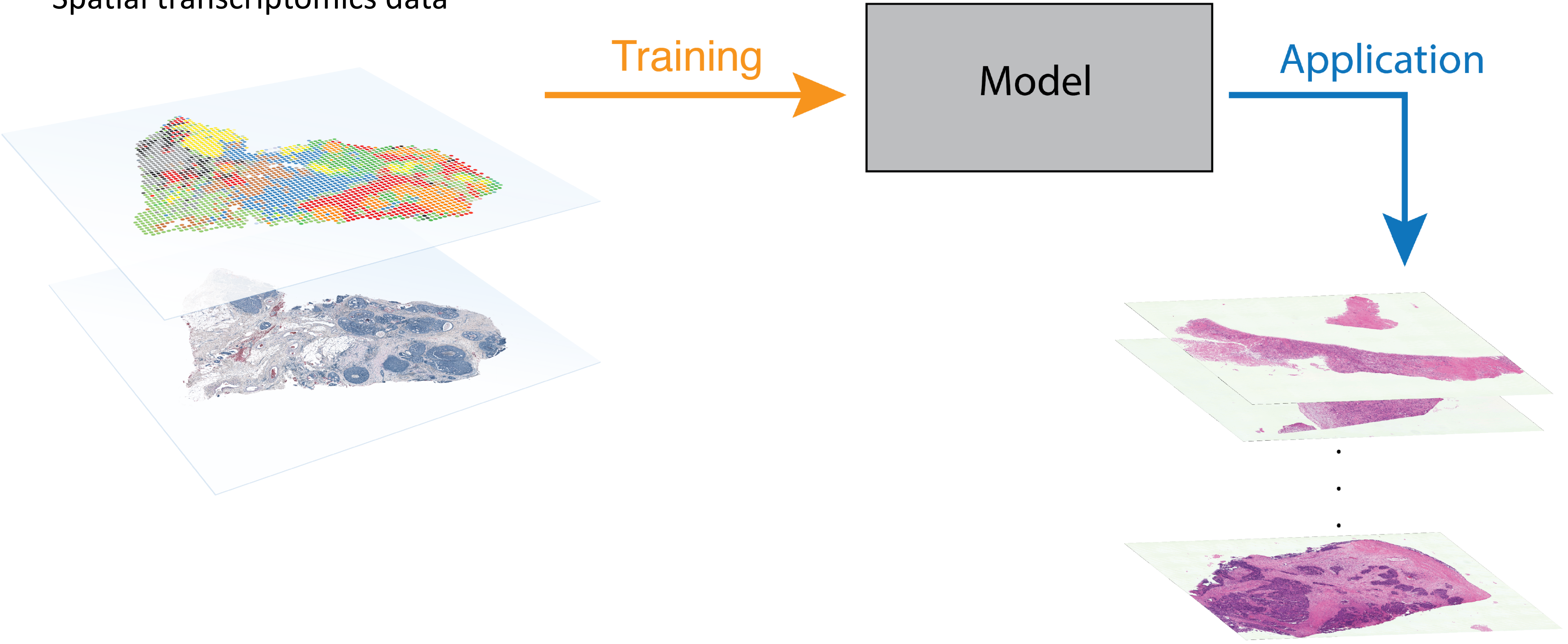




P30: pre-symptomatic
P70: onset
P100: symptomatic
P120: end-stage

Can we predict gene expression data from H&E image?

Spatial transcriptomics data



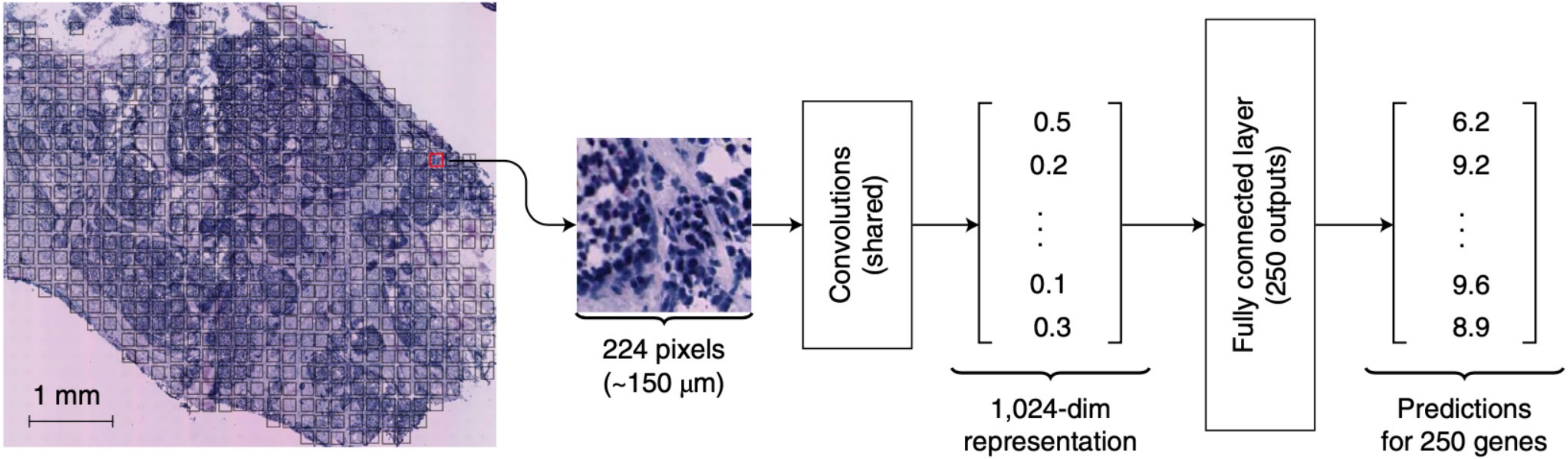
Training

Model

Application

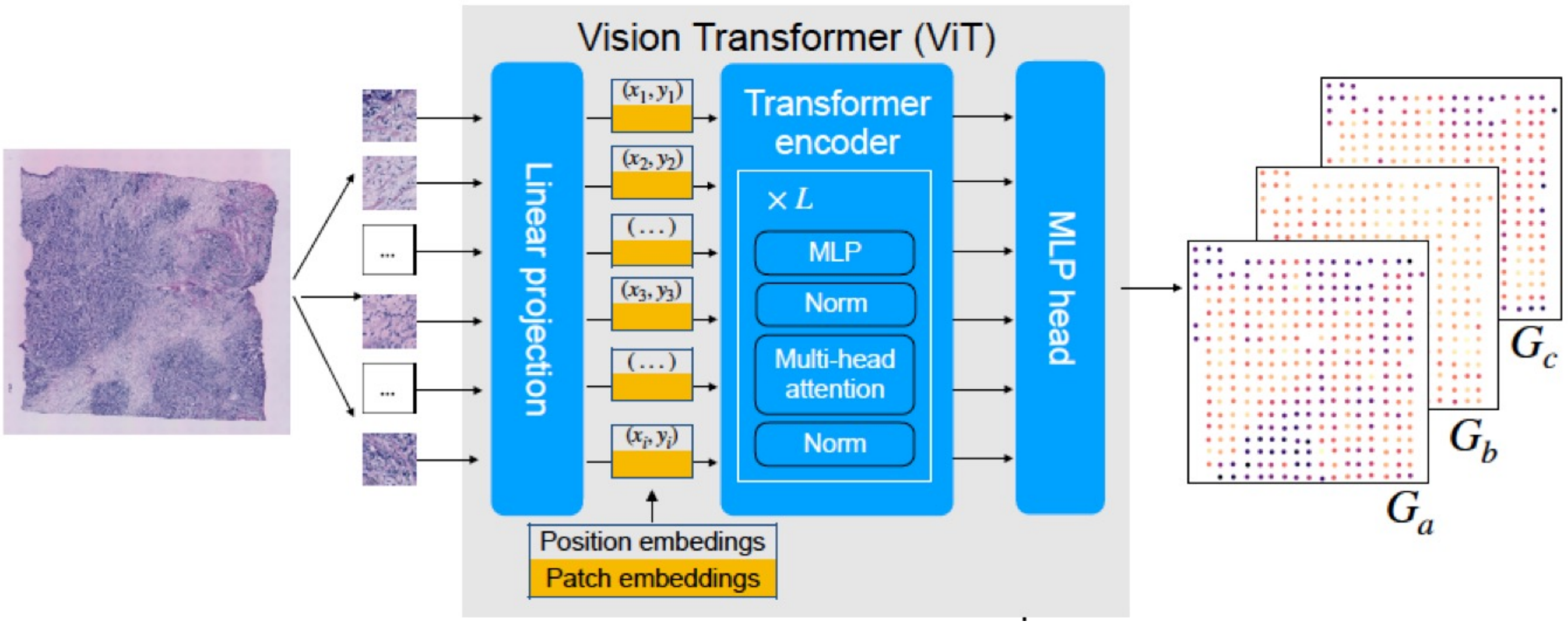
Clinical tissue slide without RNA measurement

STNet model



(He, et al., 2020)

His2genes model



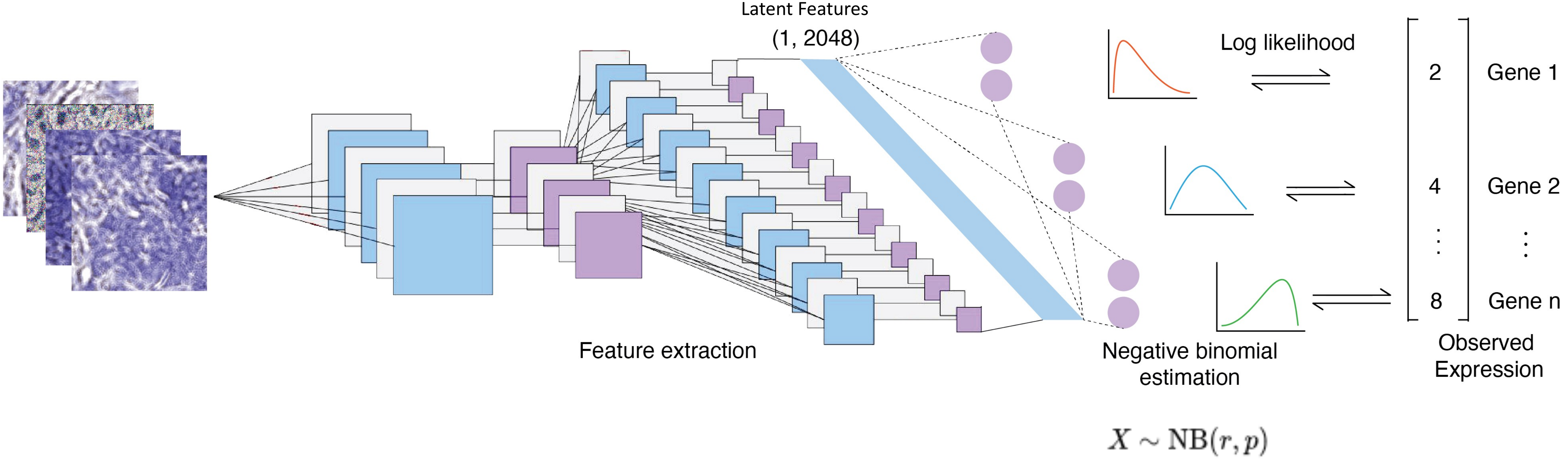
2D positional embedding

$$E = E_h + E_x + E_y.$$

$N \times 1024$	$N \times 1024$	$N \times 1024$	$N \times 1024$
	Image	X coor	Y coor

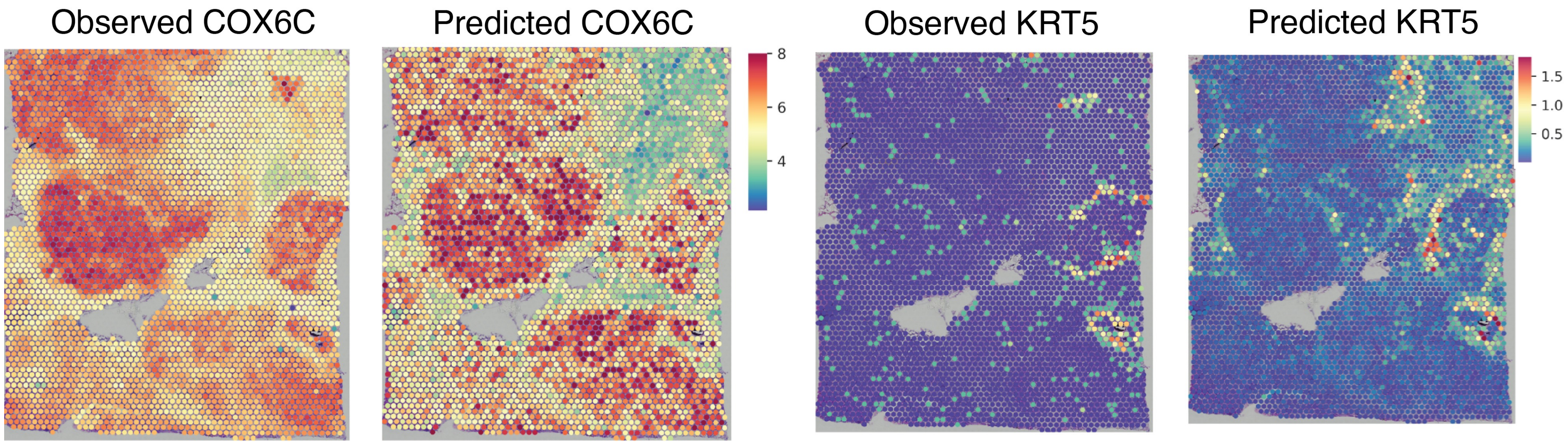
(Li, et al., 2021)

STimage: convolutional regression model



Loss: Negative log likelihood

STimage: gene expression prediction



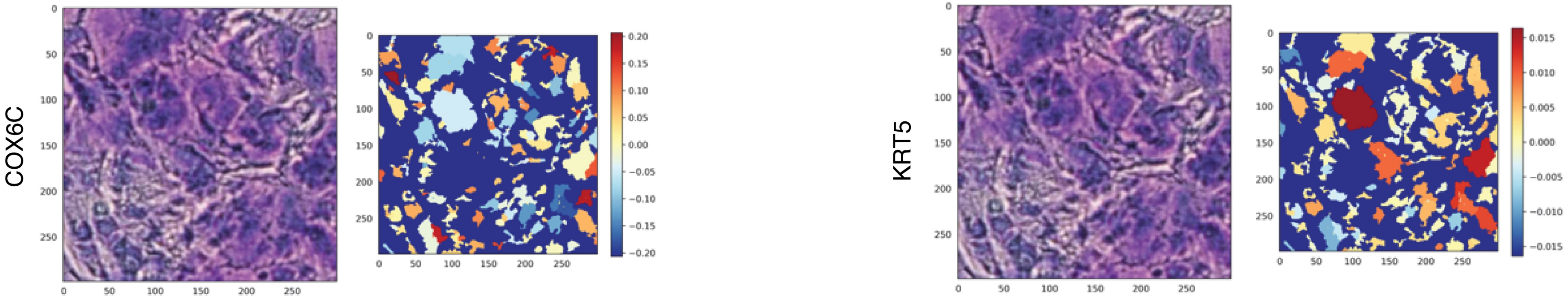
Observed COX6C

Predicted COX6C

Observed KRT5

Predicted KRT5

STimage: model interpretation

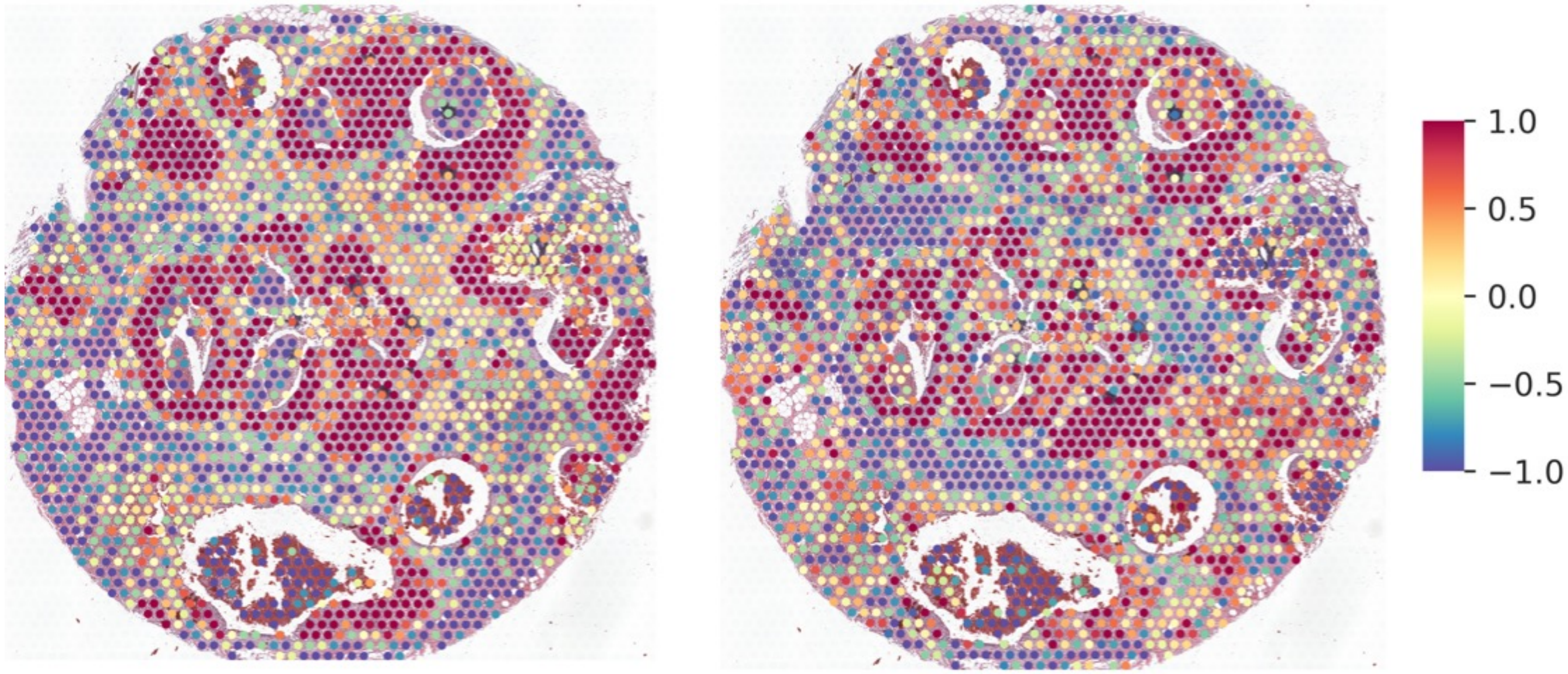


STimage: gene expression prediction on external dataset

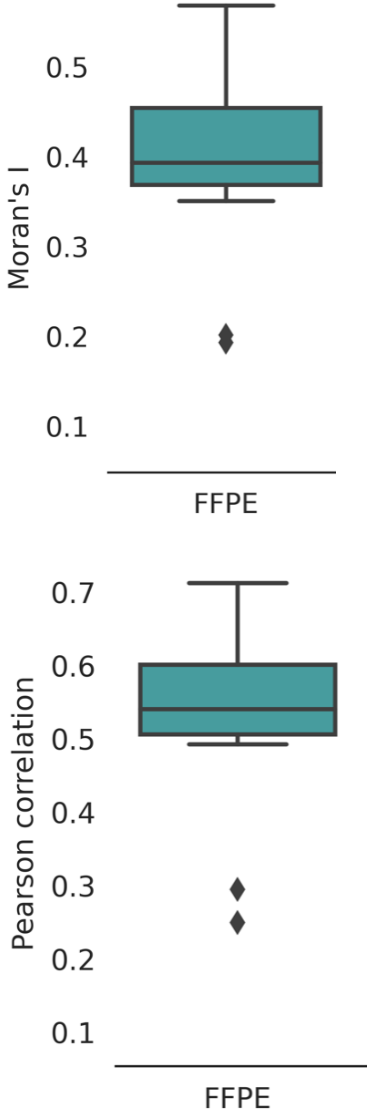
FFPE

Observed COX6C

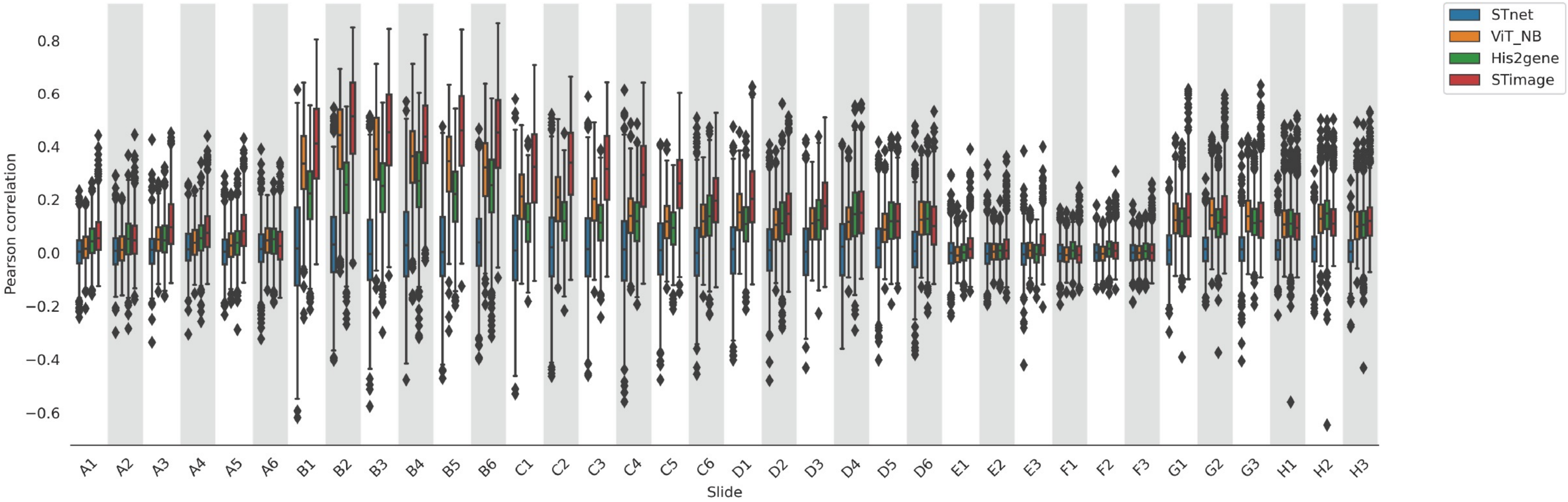
Predicted COX6C



9 breast cancer markers



Benchmarking with existing software



Interpretability Machine Learning (Deep learning)

Why

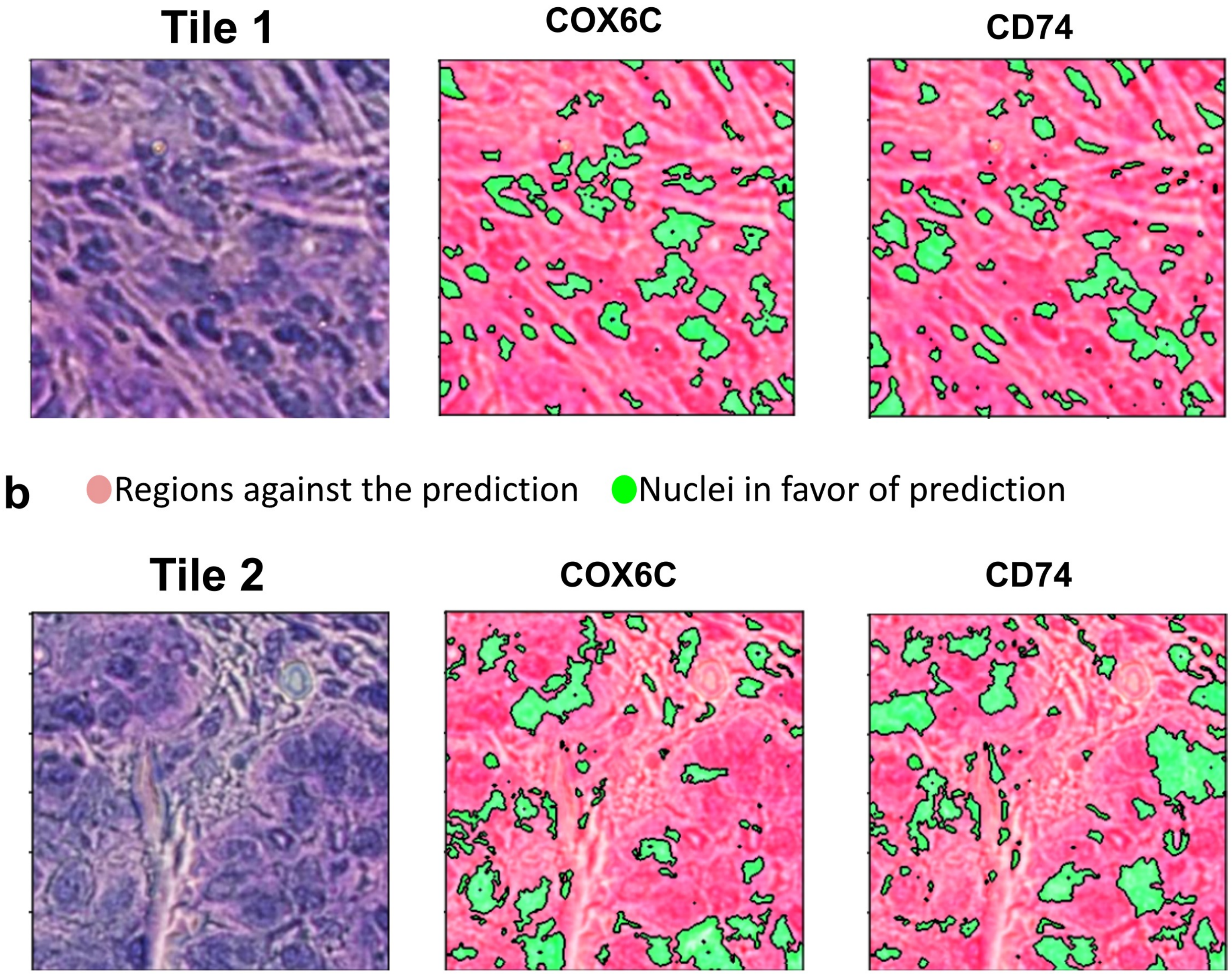
1) Bug fixing and model optimization
2) From model extracts useful information for discovery rather than performance (accuracy vs interpretability tradeoff)
3) Credibility/reliability of the model

How

1) Interpreting outputs: with saliency maps, with occlusion sensitivity, and with class activation maps (Global Average Pooling)
2) Visualisation of the model training steps: with gradient ascent (class model visualization), with dataset search, and deconvolution
3) Deep dream (going deeper in NNs) or LIME (Local interpretable model-agnostic explanations)

e.g. Saliency map compute the gradient of output category with respect to input image: $\frac{\partial output}{\partial input}$

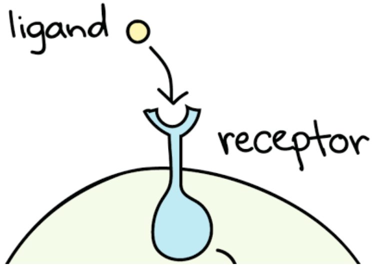
Interpretability Machine Learning (Deep learning)

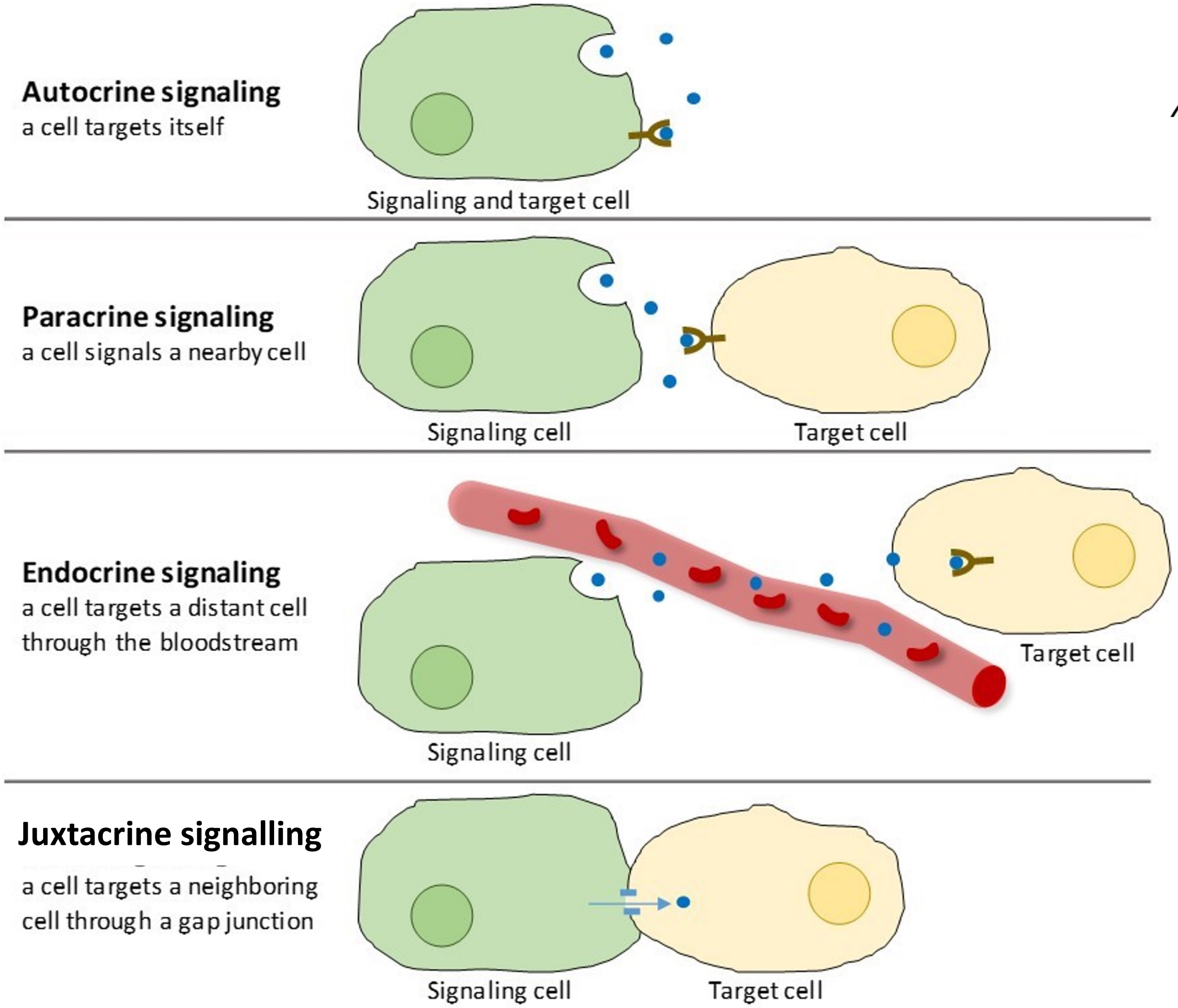

Tile 1
COX6C
CD74
b
● Regions against the prediction
● Nuclei in favor of prediction
Tile 2
COX6C
CD74

LIME uses perturbations to find those segments of the image which are more predictive of high or low expression across an image.

Analysis of Cell-Cell Interactions

Cell-to-cell interaction/communication concept





Autocrine signaling
a cell targets itself

Paracrine signaling
a cell signals a nearby cell

Endocrine signaling
a cell targets a distant cell through the bloodstream

Juxtacrine signalling
a cell targets a neighboring cell through a gap junction

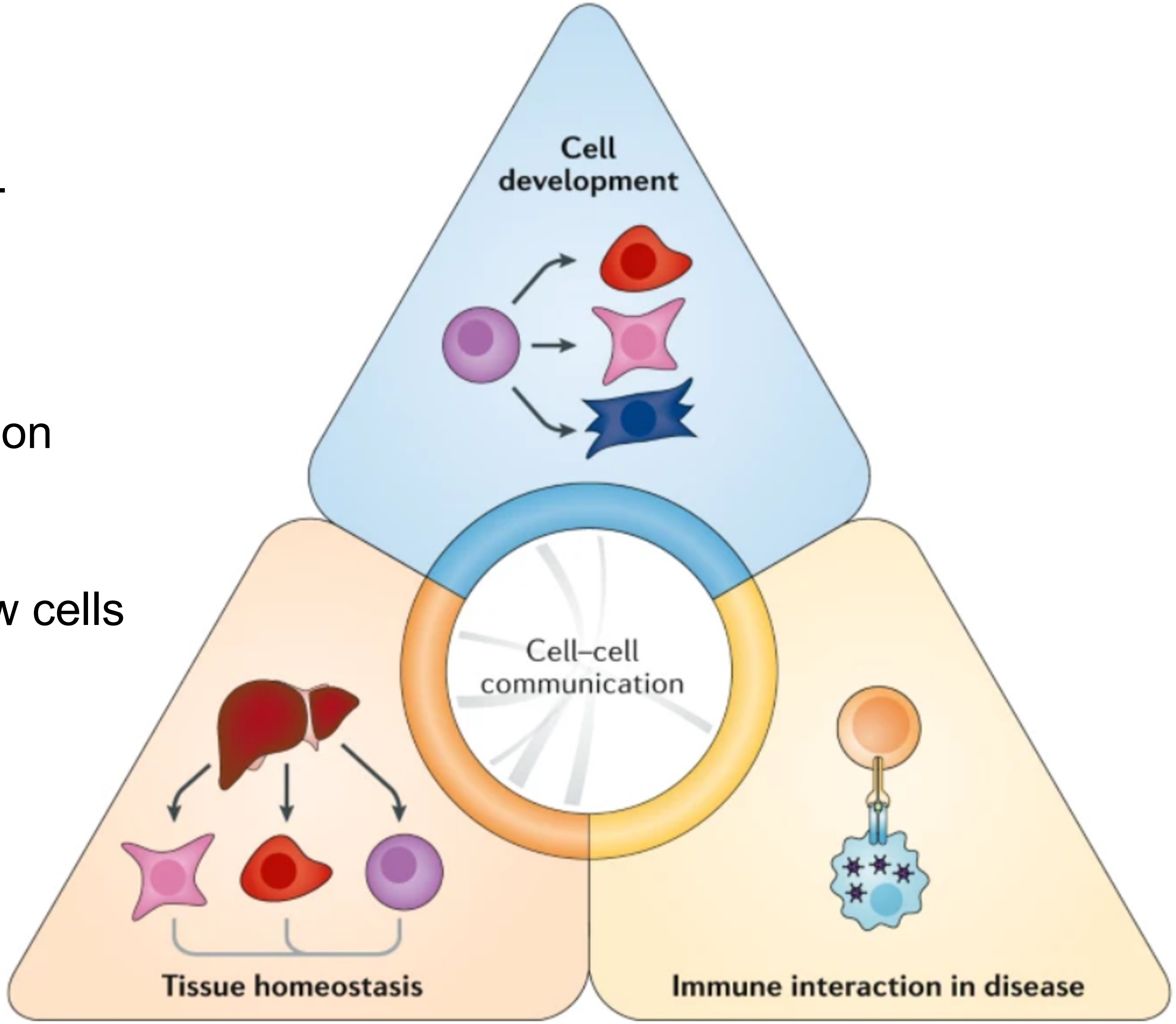
Application of cell-cell interaction (CCI) analysis

Examples of application:

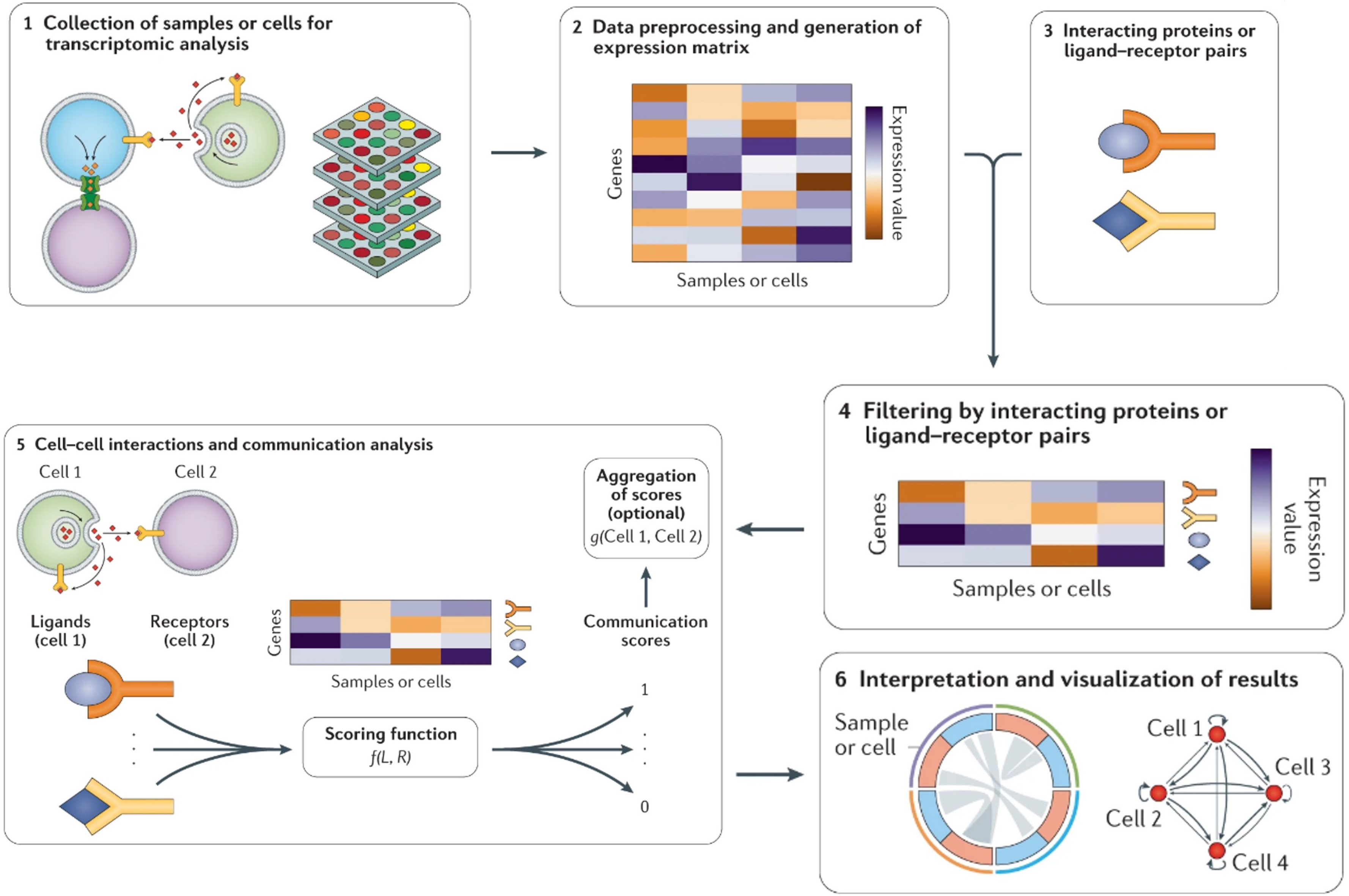
Cell development:
Revealed ligand–receptor interactions that initiate self-renewal and differentiation

Tissue homeostasis:
Intercellular communication contributes to organ function

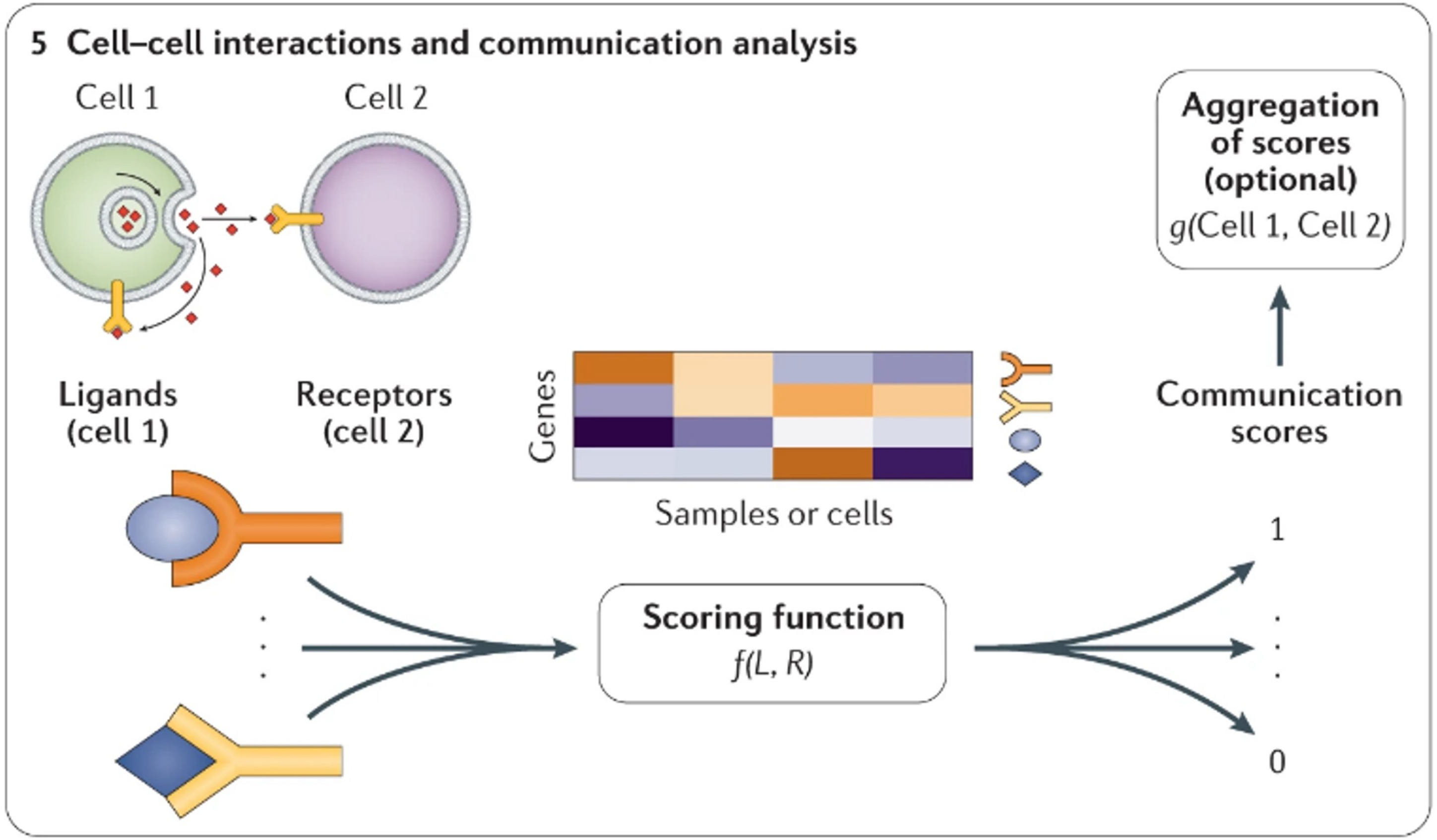
Immune interaction in disease:
Studying CCI within these communities can reveal how cells communicate in these ecosystems and help guide the development of effective cancer immunotherapies



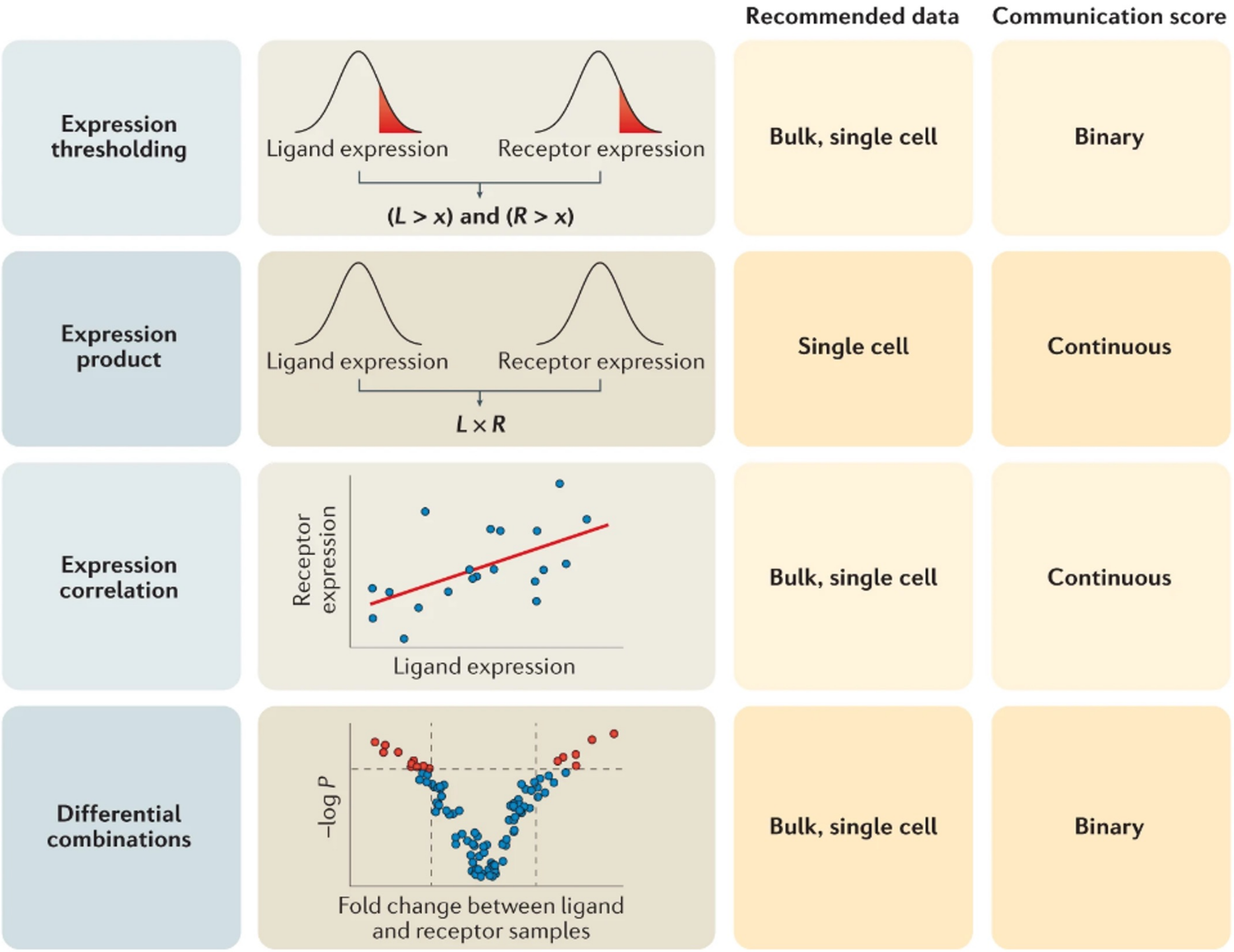
Basic workflow of CCI analysis with transcriptomics data


1 Collection of samples or cells for transcriptomic analysis
2 Data preprocessing and generation of expression matrix
Genes
Expression value
Samples or cells
3 Interacting proteins or ligand–receptor pairs
4 Filtering by interacting proteins or ligand–receptor pairs
Genes
Samples or cells
Expression value
5 Cell–cell interactions and communication analysis
Cell 1
Cell 2
Ligands (cell 1)
Receptors (cell 2)
Genes
Samples or cells
Aggregation of scores (optional)
g(Cell 1, Cell 2)
Communication scores
Scoring function
f(L, R)
1
0
6 Interpretation and visualization of results
Sample or cell
Cell 1
Cell 2
Cell 3
Cell 4

General method

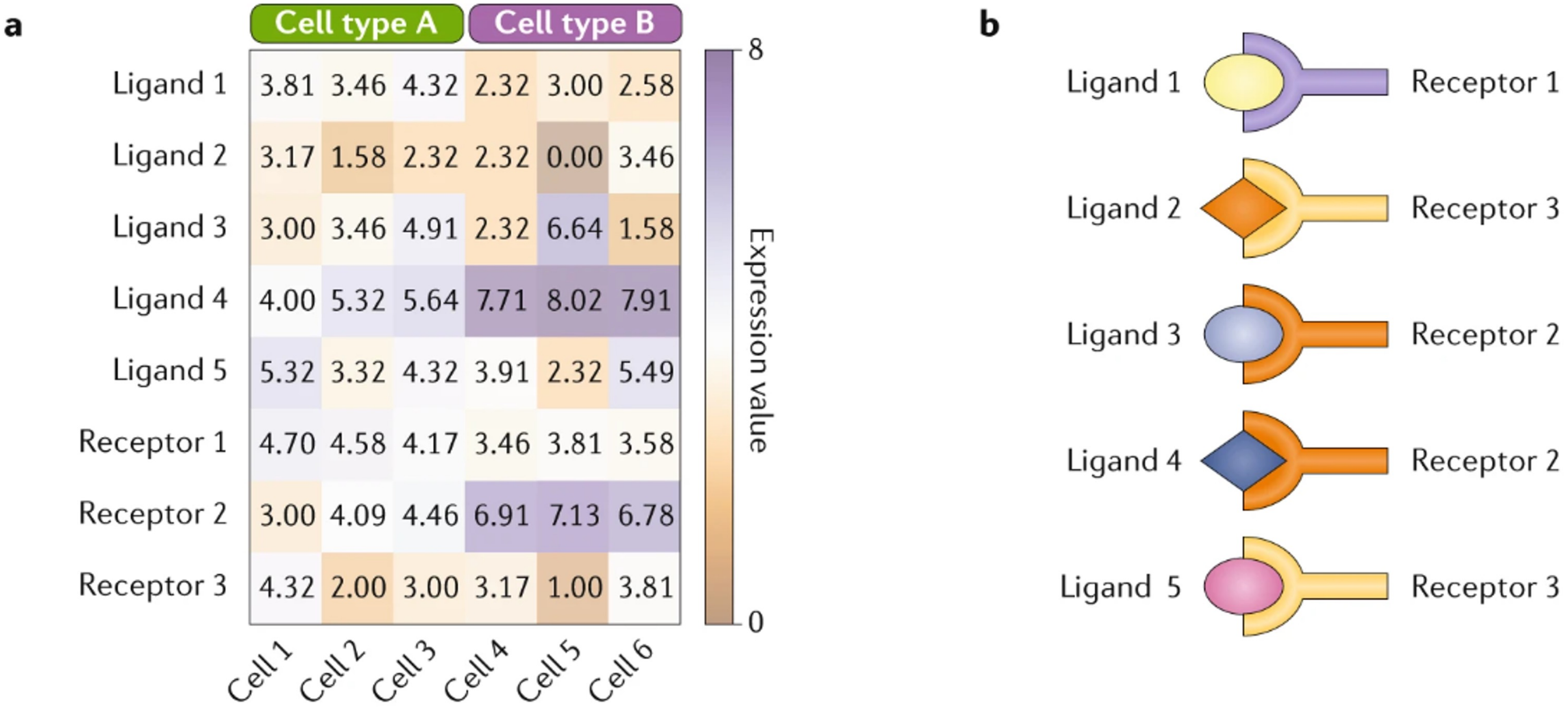


Main scoring functions with gene expression data



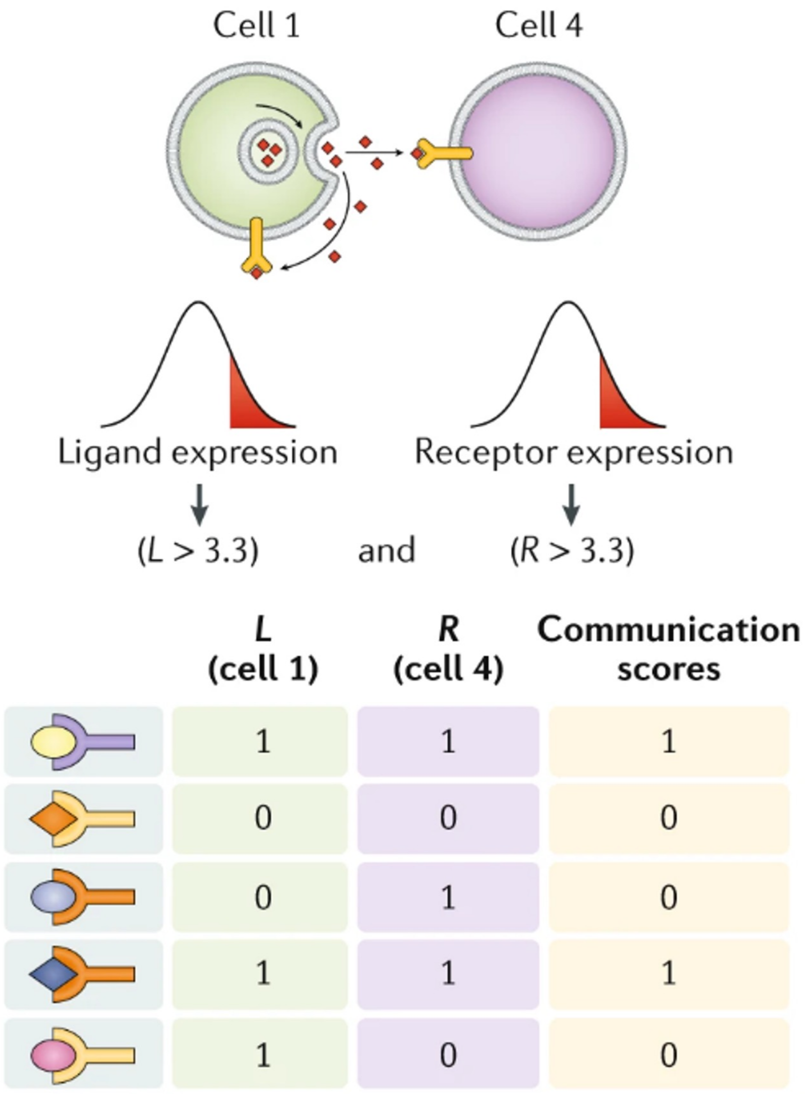
		Recommended data	Communication score
Expression thresholding	Ligand expression Receptor expression $(L > x)$ and $(R > x)$	Bulk, single cell	Binary
Expression product	Ligand expression Receptor expression $L \times R$	Single cell	Continuous
Expression correlation	Receptor expression Ligand expression	Bulk, single cell	Continuous
Differential combinations	$-\log P$ Fold change between ligand and receptor samples	Bulk, single cell	Binary

Toy example



	Cell type A			Cell type B		
	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6
Ligand 1	3.81	3.46	4.32	2.32	3.00	2.58
Ligand 2	3.17	1.58	2.32	2.32	0.00	3.46
Ligand 3	3.00	3.46	4.91	2.32	6.64	1.58
Ligand 4	4.00	5.32	5.64	7.71	8.02	7.91
Ligand 5	5.32	3.32	4.32	3.91	2.32	5.49
Receptor 1	4.70	4.58	4.17	3.46	3.81	3.58
Receptor 2	3.00	4.09	4.46	6.91	7.13	6.78
Receptor 3	4.32	2.00	3.00	3.17	1.00	3.81

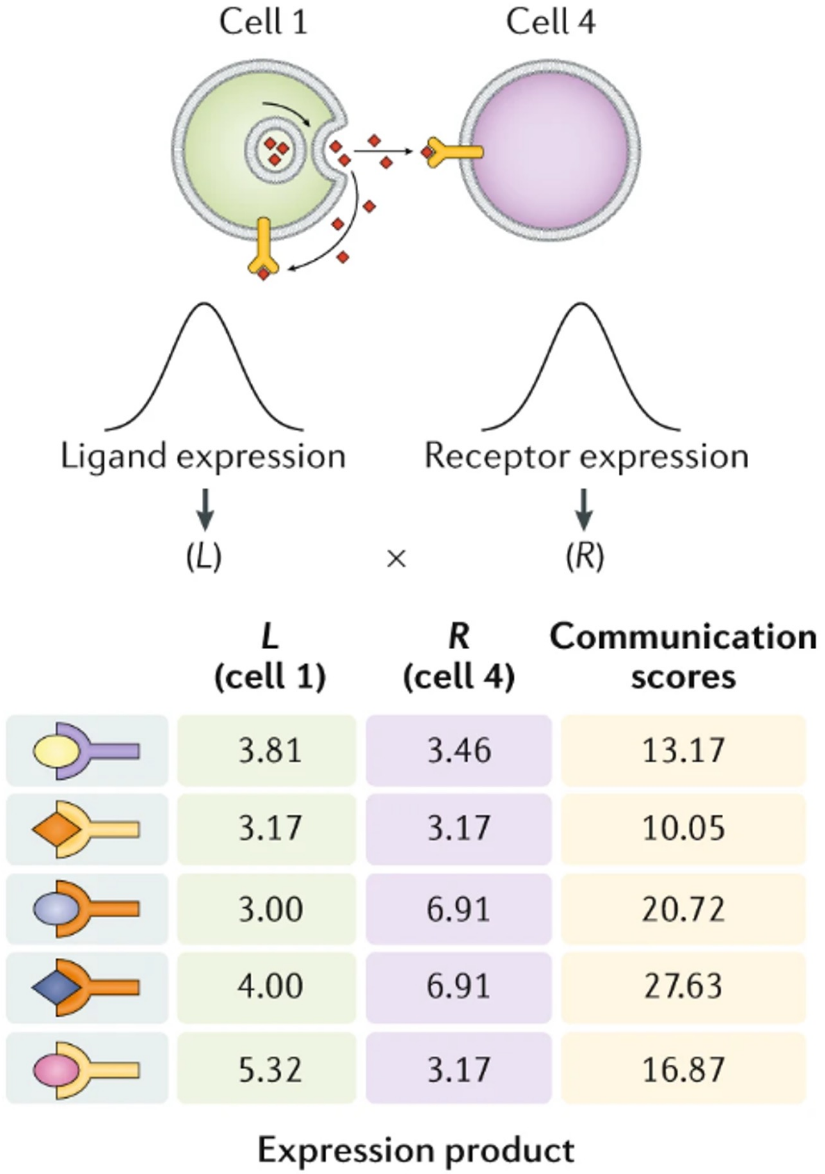
Expression thresholding



	L **(cell 1)**	***R*** **(cell 4)**	**Communication scores**
	1	1	1
	0	0	0
	0	1	0
	1	1	1
	1	0	0

Expression thresholding

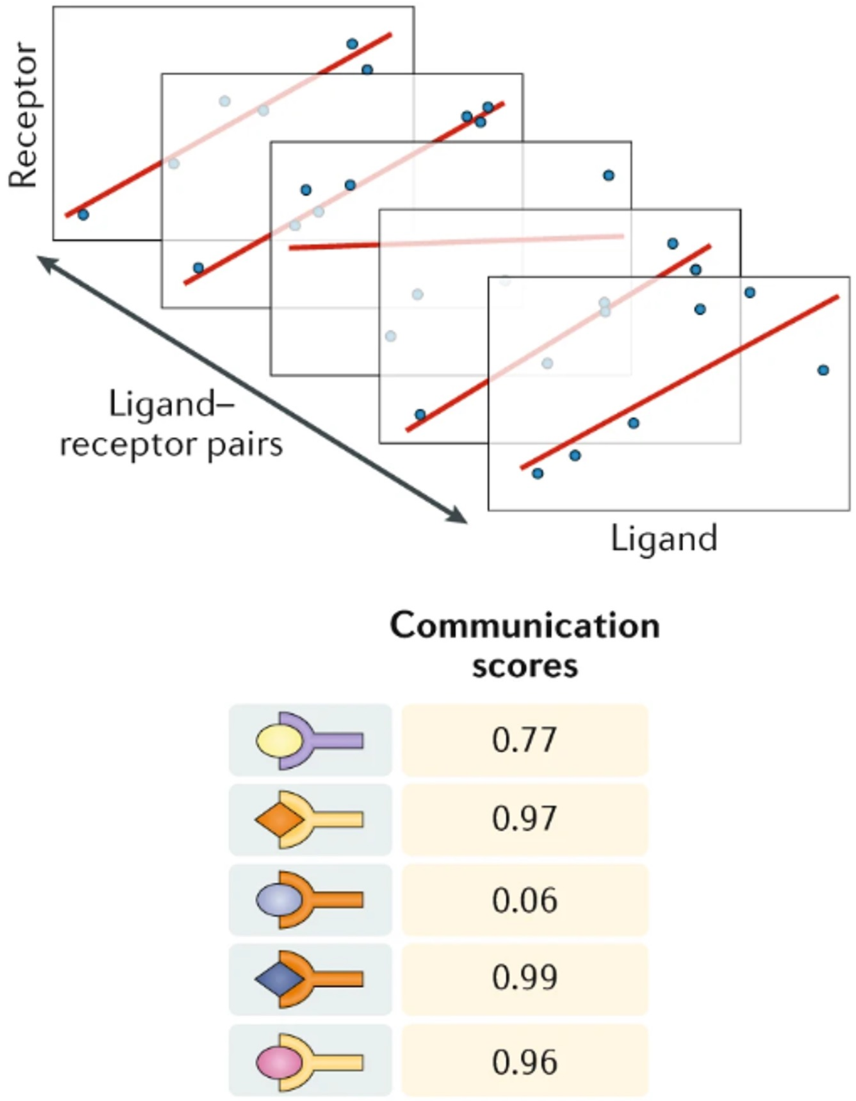
Expression product



	L **(cell 1)**	***R*** **(cell 4)**	**Communication scores**
	3.81	3.46	13.17
	3.17	3.17	10.05
	3.00	6.91	20.72
	4.00	6.91	27.63
	5.32	3.17	16.87

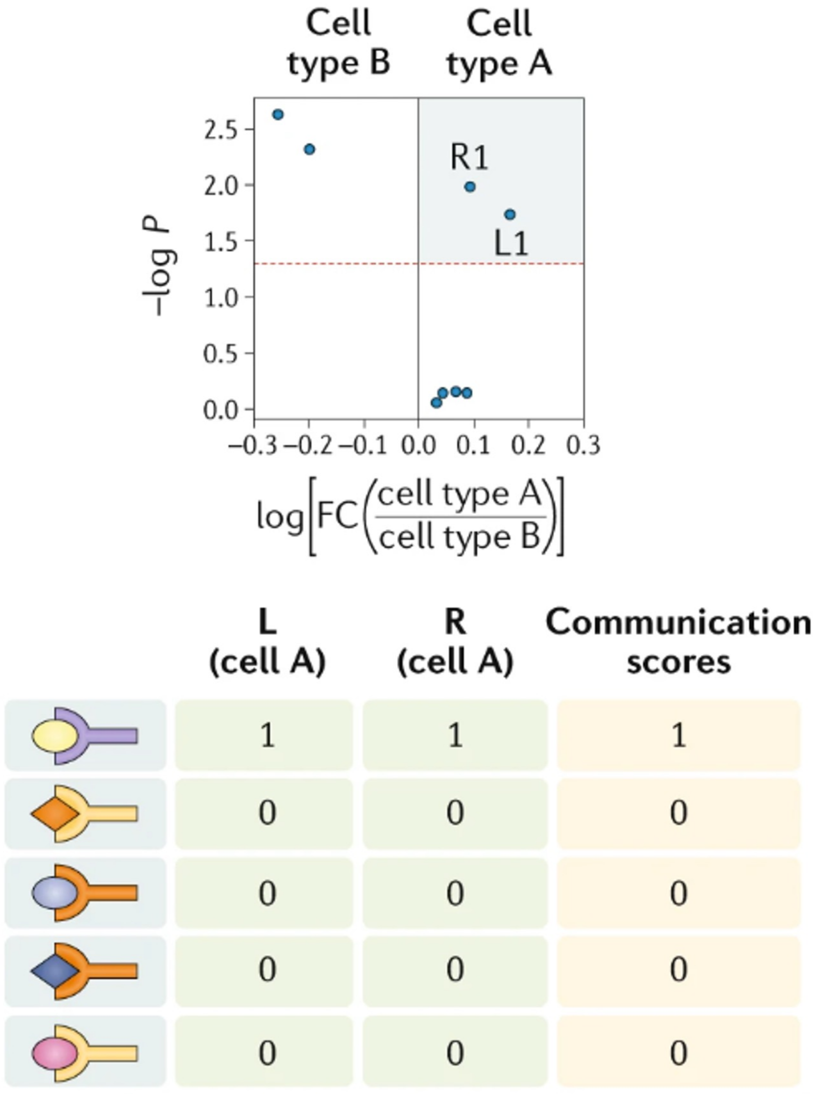
Expression product

Expression correlation



Expression correlation

Differential combinations



	L (cell A)	R (cell A)	Communication scores
	1	1	1
	0	0	0
	0	0	0
	0	0	0
	0	0	0

Differential combinations

Spatial context in CCI analysis

scRNAseq

False positive CCI ⬆

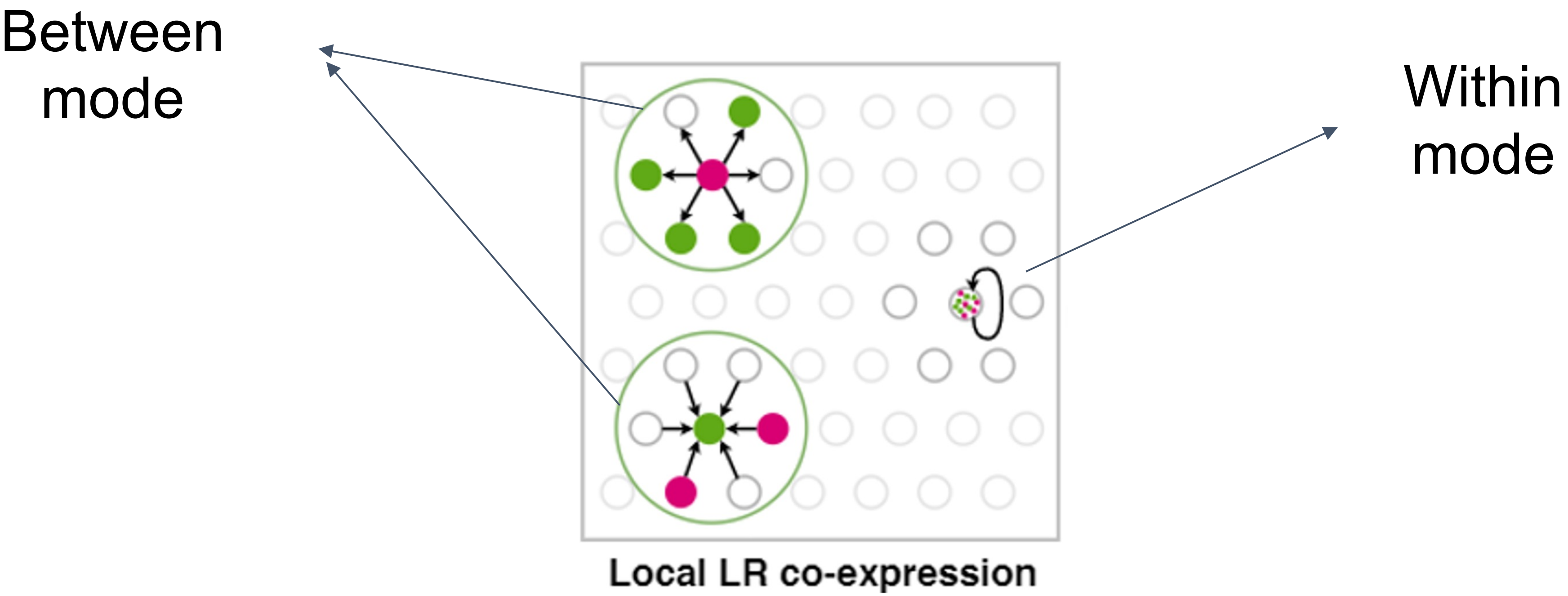
- Missing spatial contact information
- High false-positive CCI prediction

Spatial transcriptomics

Clarity ⬆

- Cell localization can help elucidate interactions between spatially proximal regions.

Expression product with neighborhood score



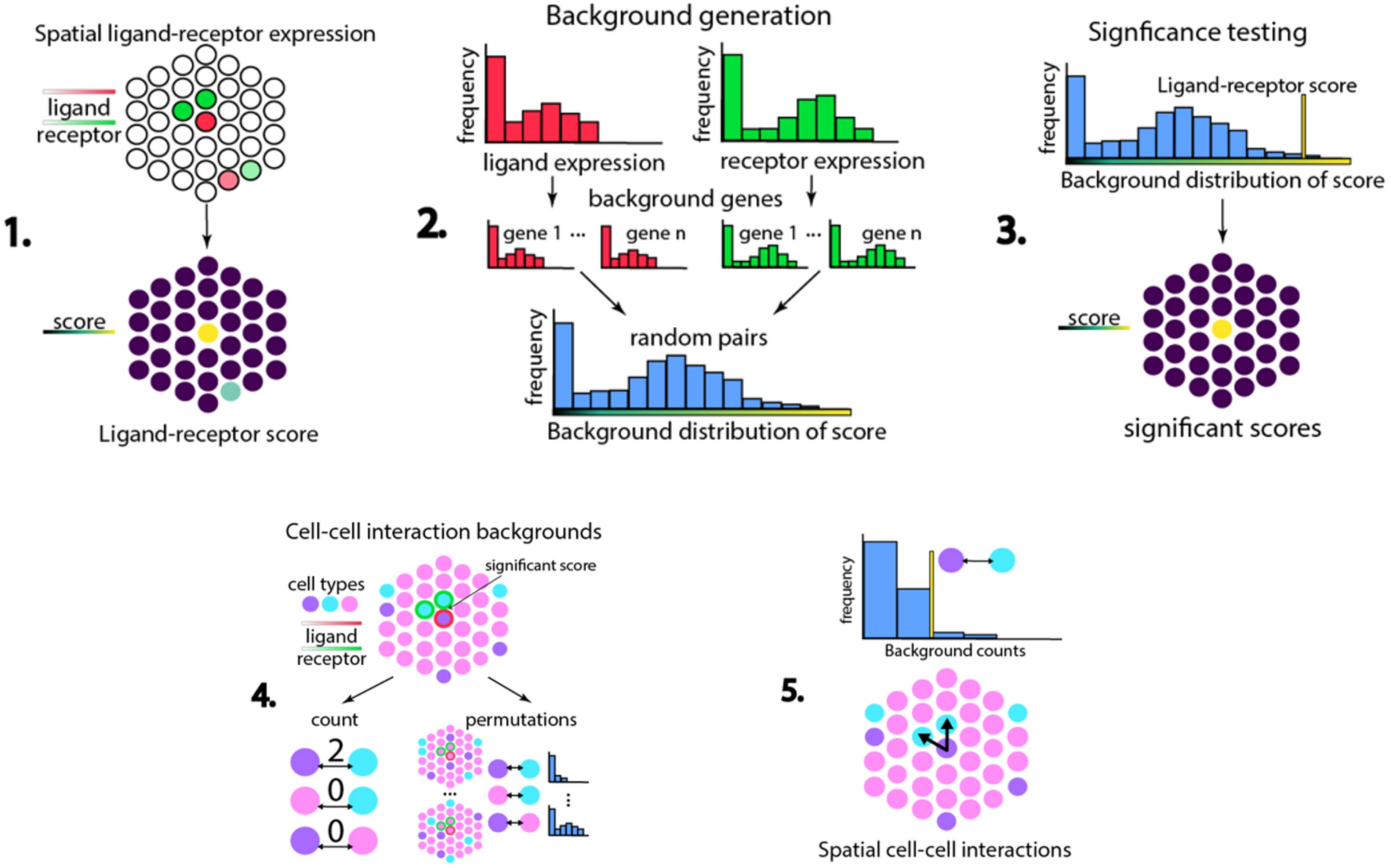
Between mode

Within mode

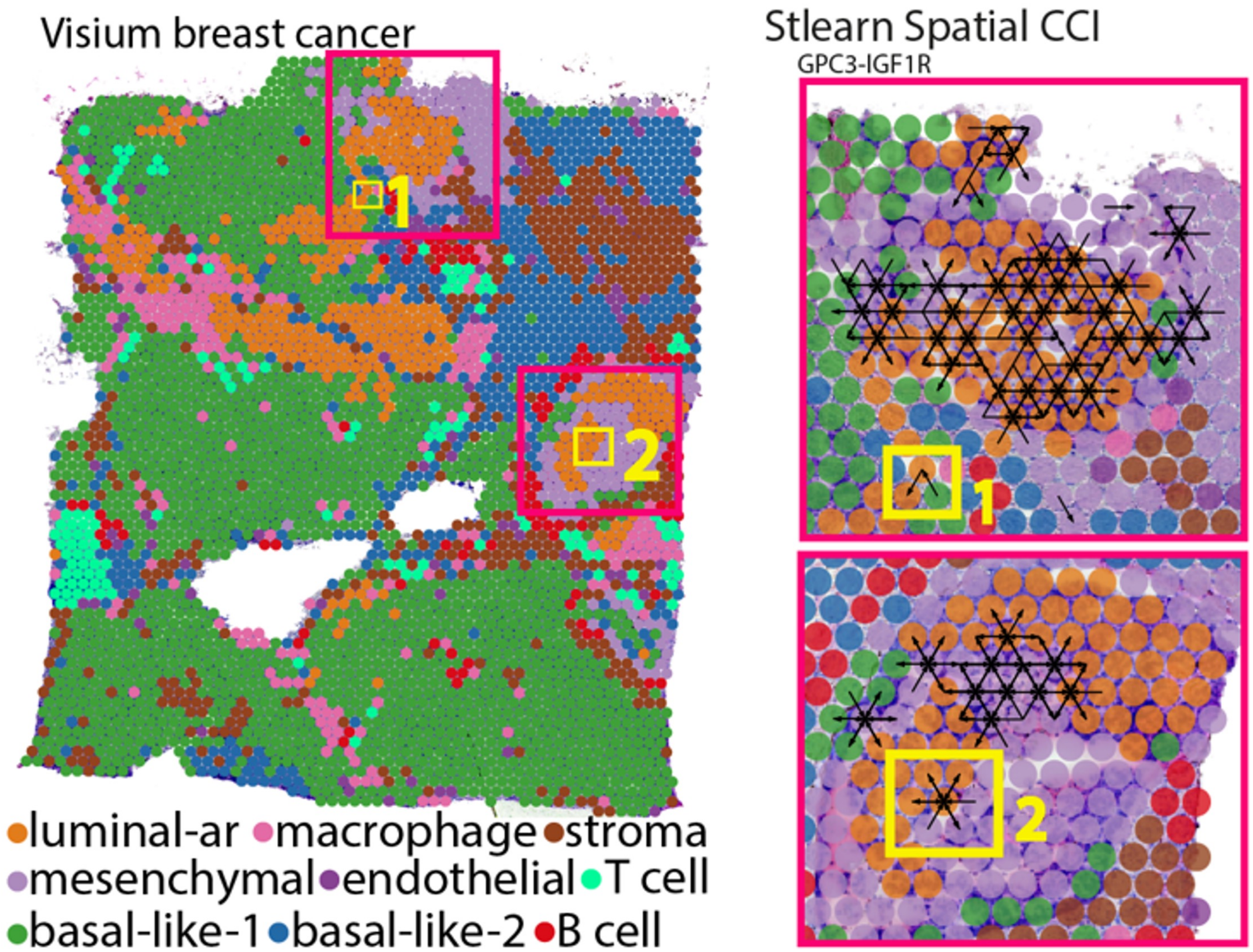
Local LR co-expression

$$LR_{score} = \frac{1}{2}(mean(Expr_{L,S|N} \times [Expr_{R,S} > 0])$$
$$+ mean(Expr_{R,S|N} \times [Expr_{L,S} > 0])$$

Spatial CCI with significant testing



Example: Immune interaction Breast cancer


Visium breast cancer
1
2
●luminal-ar ●macrophage ●stroma
●mesenchymal ●endothelial ●T cell
●basal-like-1 ●basal-like-2 ●B cell
Stlearn Spatial CCI
GPC3-IGF1R
1
2

Discussion and Future Perspectives