Module 2 Practical Day 2

Learning Objectives

- Understand new biology learnt from spatial community analyses
- Integrate spatial transcriptomics with GWAS summary statistics
- Understand neural network (NN) architectures:
  - Autoencoders
  - CNN
  - Transformers
- How to apply NN on single cell and spatial data

Module 2 - running the learning materials

https://github.com/GenomicsMachineLearning/gml-teaching-2026

Copy and paste each of the following lines into your terminal once you have logged into the workshop server:

- `/software/bin/micromamba shell init`
- `source ~/.bashrc`
- `micromamba activate /software/conda-envs/gml-teaching`
- `git clone https://github.com/GenomicsMachineLearning/gml-teaching-2026`
- `~/qimr-teaching-2026/runme.sh`

The output will look something like:

```
Port 3502 is available

Command to create ssh tunnel:
ssh -N -L 3502:10.10.10.10:3502 foo@10.10.10.10
Use a Browser on your local machine to go to:
localhost:3502  (prefix w/ https:// if using password)

[I 2024-06-20 05:57:41.633 ServerApp] Extension package jupyter_lsp took 0.1372s to import
[I 2024-06-20 05:57:44.647 ServerApp]     http://127.0.0.1:3502/tree?token=abc123
```

- Copy the line beginning with "ssh" into a new terminal, on your local computer, and hit [Enter].
- Copy the text beginning with "http://127.0.0.1" into a new tab in your browser, and hit [Enter].

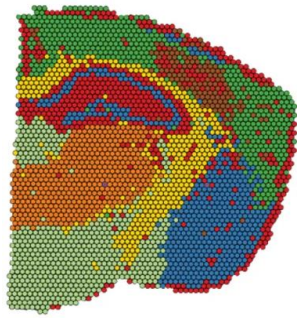
Practical 3: Spatial community analysis

- Prakrithi Pavithra

TensorFlow

PyTorch

Python packages



10X Visium

Running the Practical

Terminal

PowerShell

1. Log into your account:

   ssh {username}@203.101.229.126
   username & password from winter school email

2. Follow these commands:

   - /software/bin/micromamba shell init
   - source ~/.bashrc
   - micromamba activate /software/conda-envs/winter_school_2026
   - git clone https://github.com/GenomicsMachineLearning/gml-teaching-2026
   - cd /scratch/$USER/gml-teaching-2025
   - /scratch/$USER/gml-teaching-2025/runme.sh

3. Open JuperterNotebook:

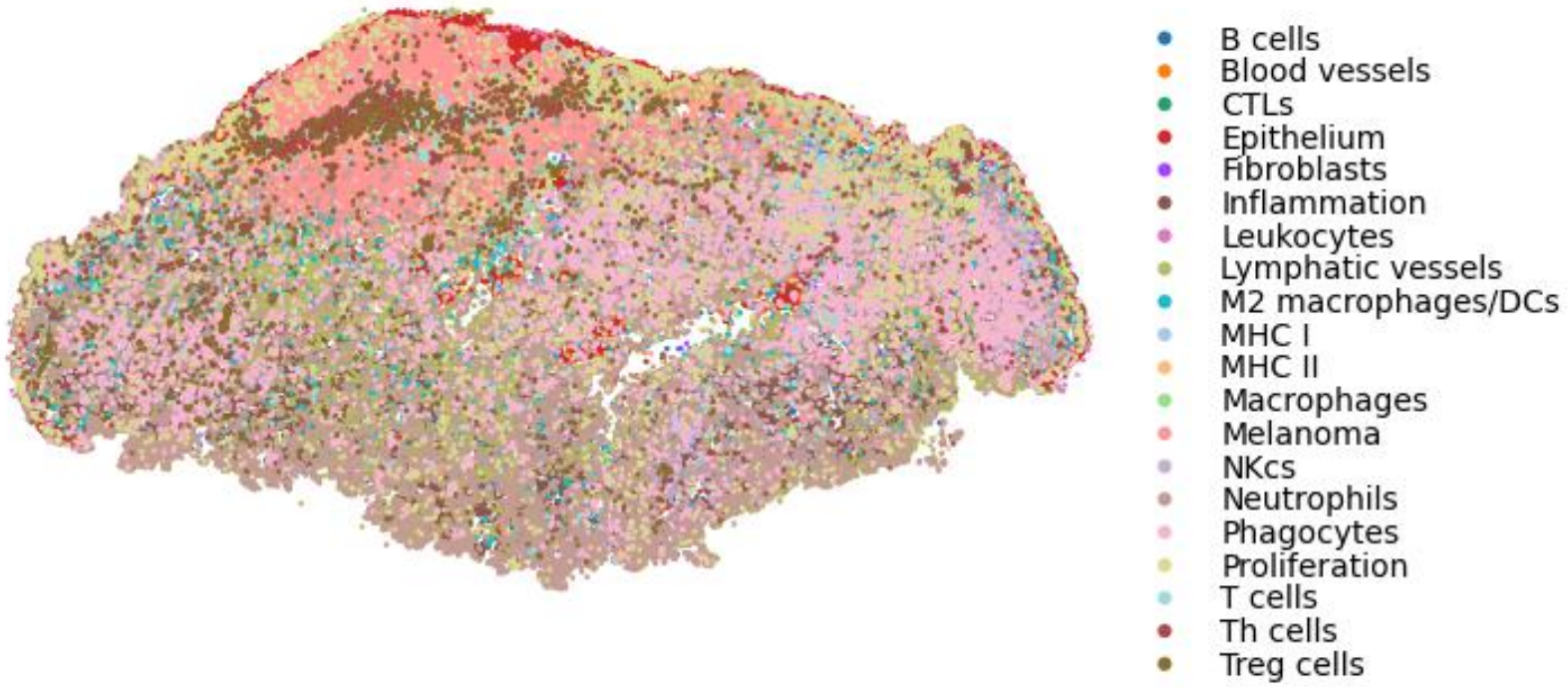
   003-downstream-analysis

Community analysis

hoodscanR

A user-friendly R package designed explicitly for cellular neighborhood analysis at single-cell resolution.

Cell type annotation



Marker list

…

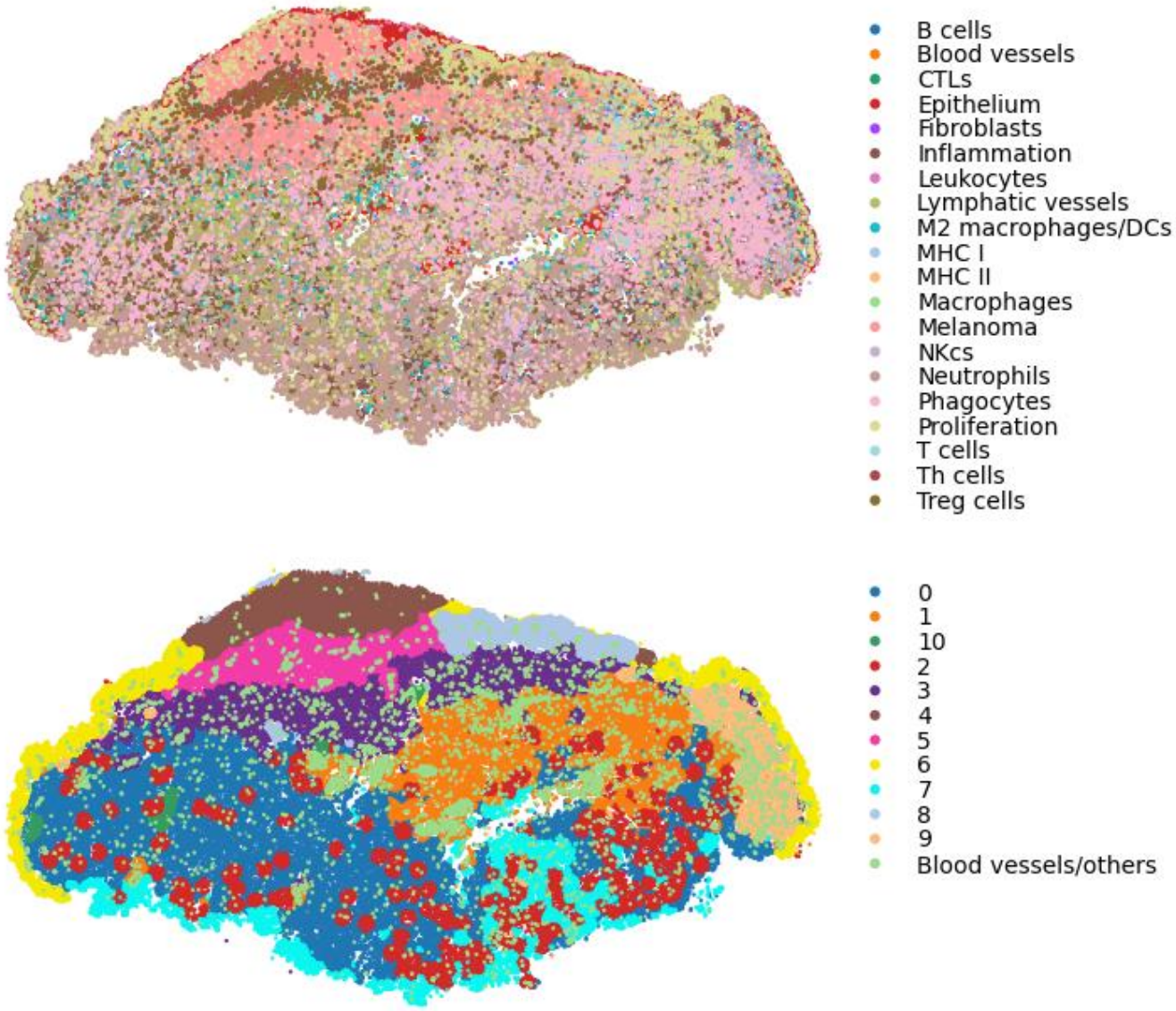
'Keratin 14': 'Epithelium',

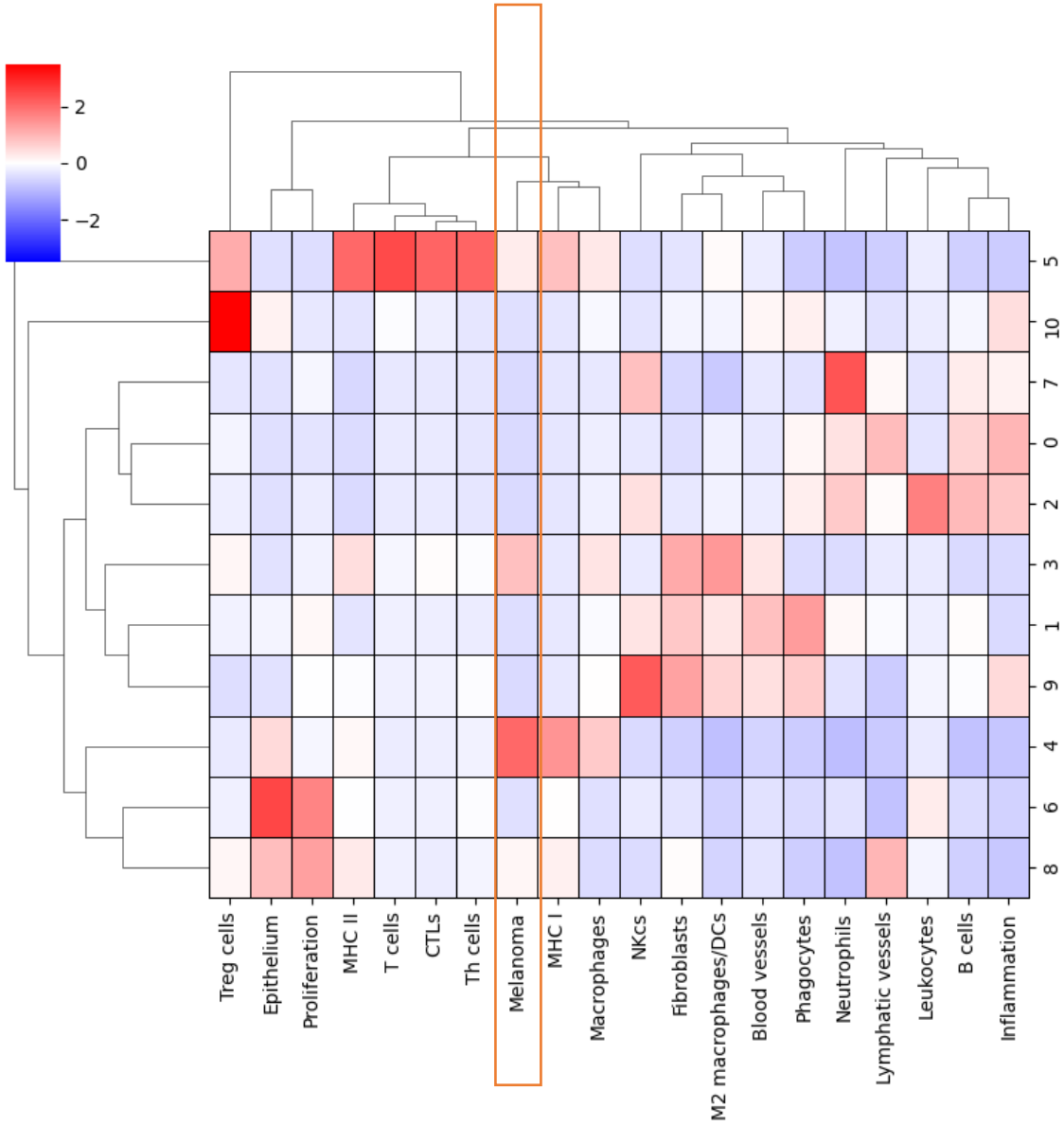
'Ki67': 'Proliferation',

'S100B': 'Melanoma',

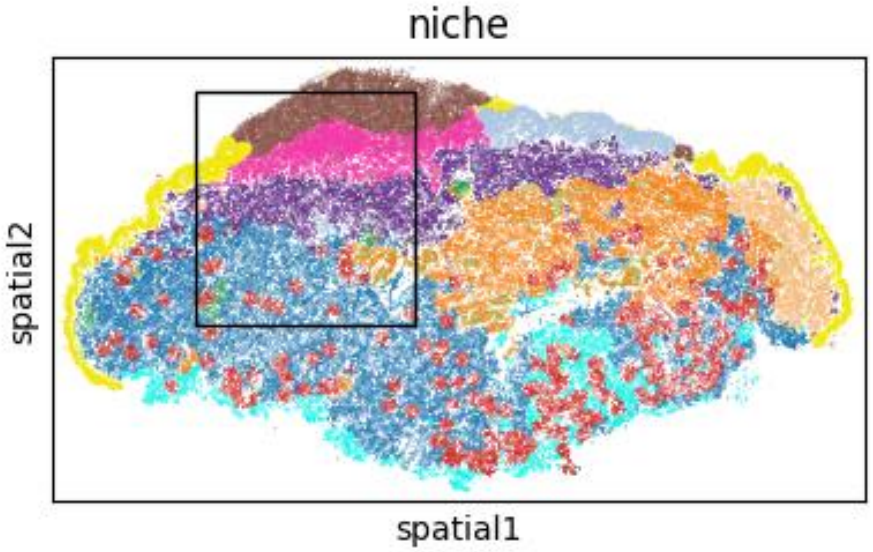
…

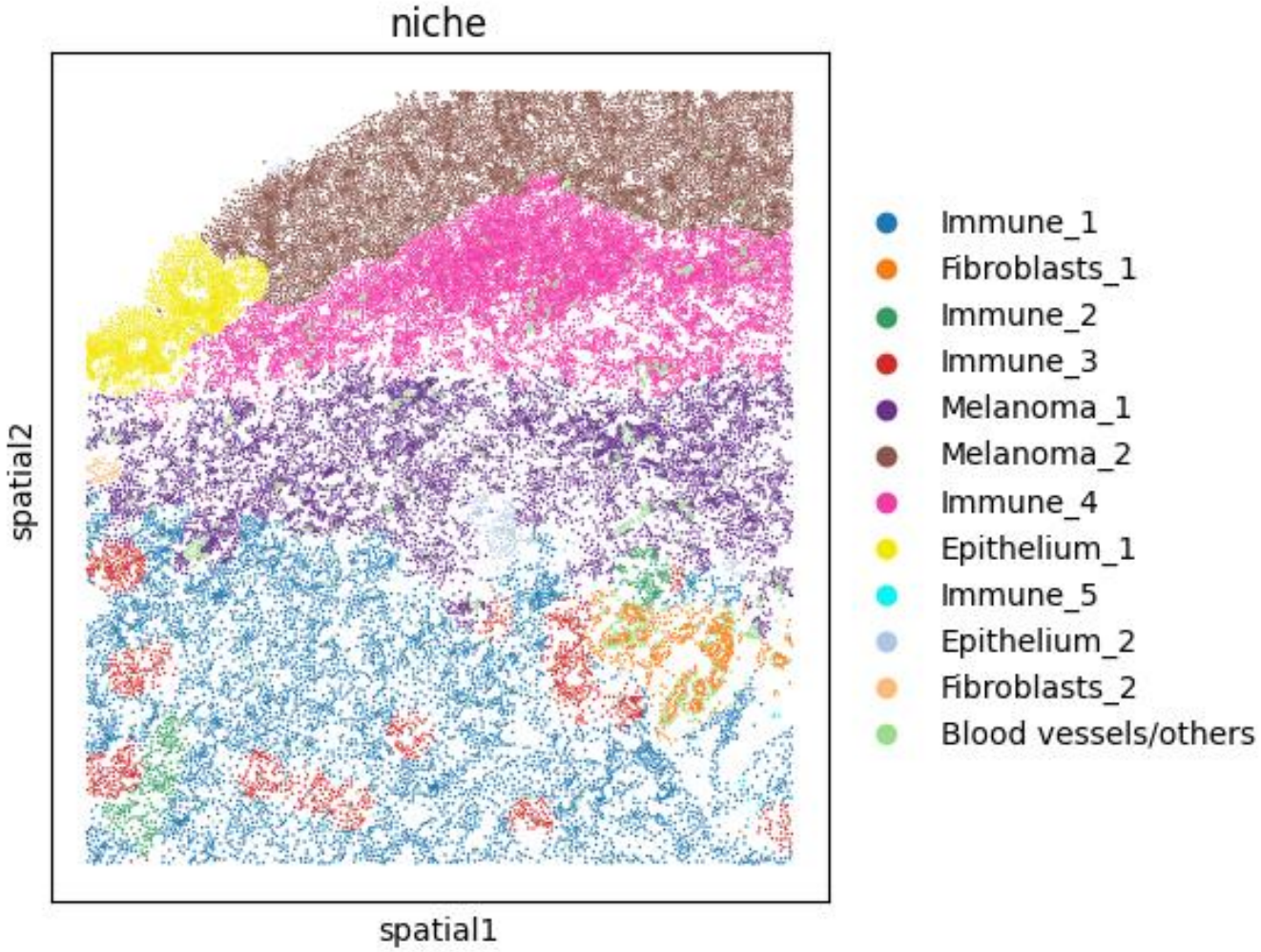
Cell type niche



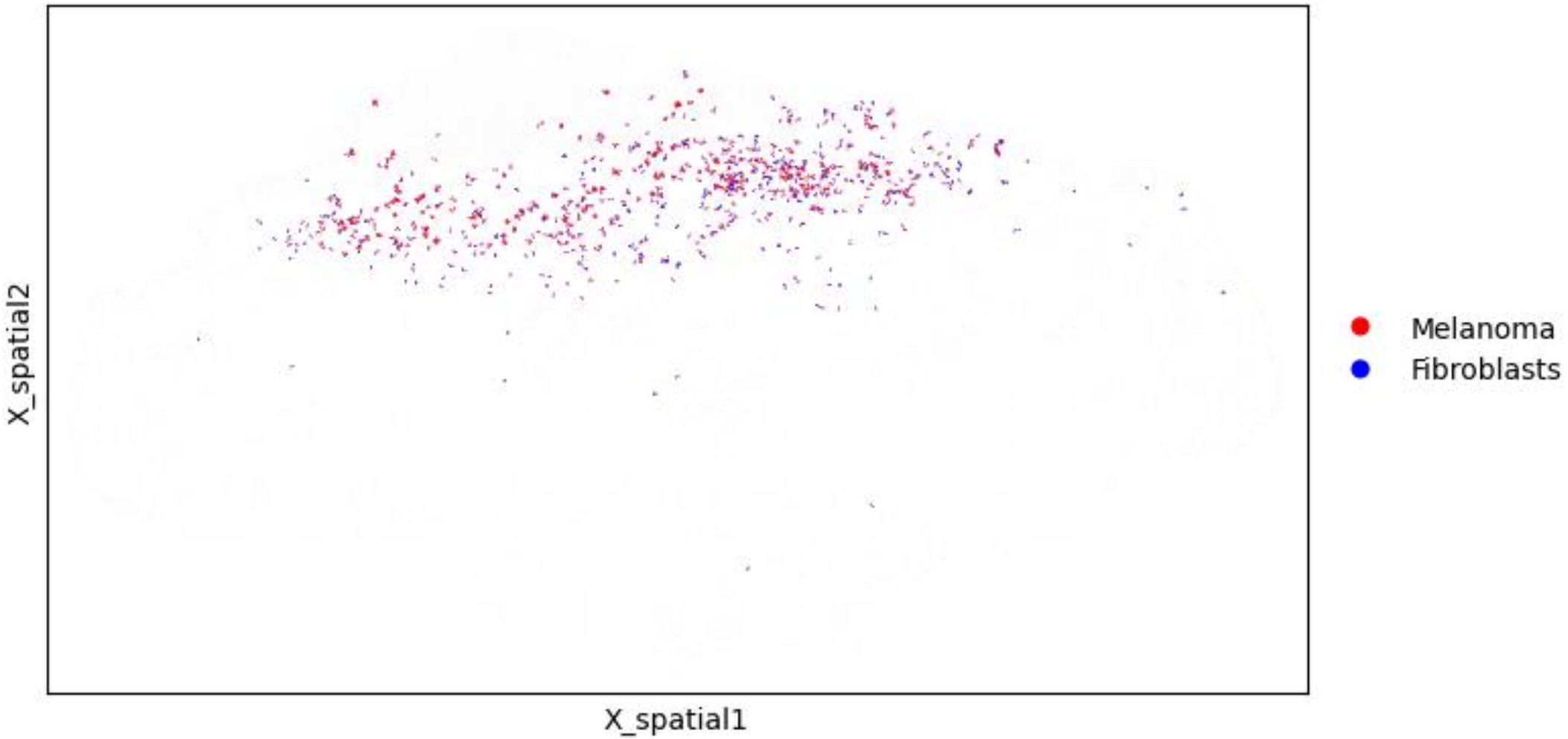


Cell type niche components

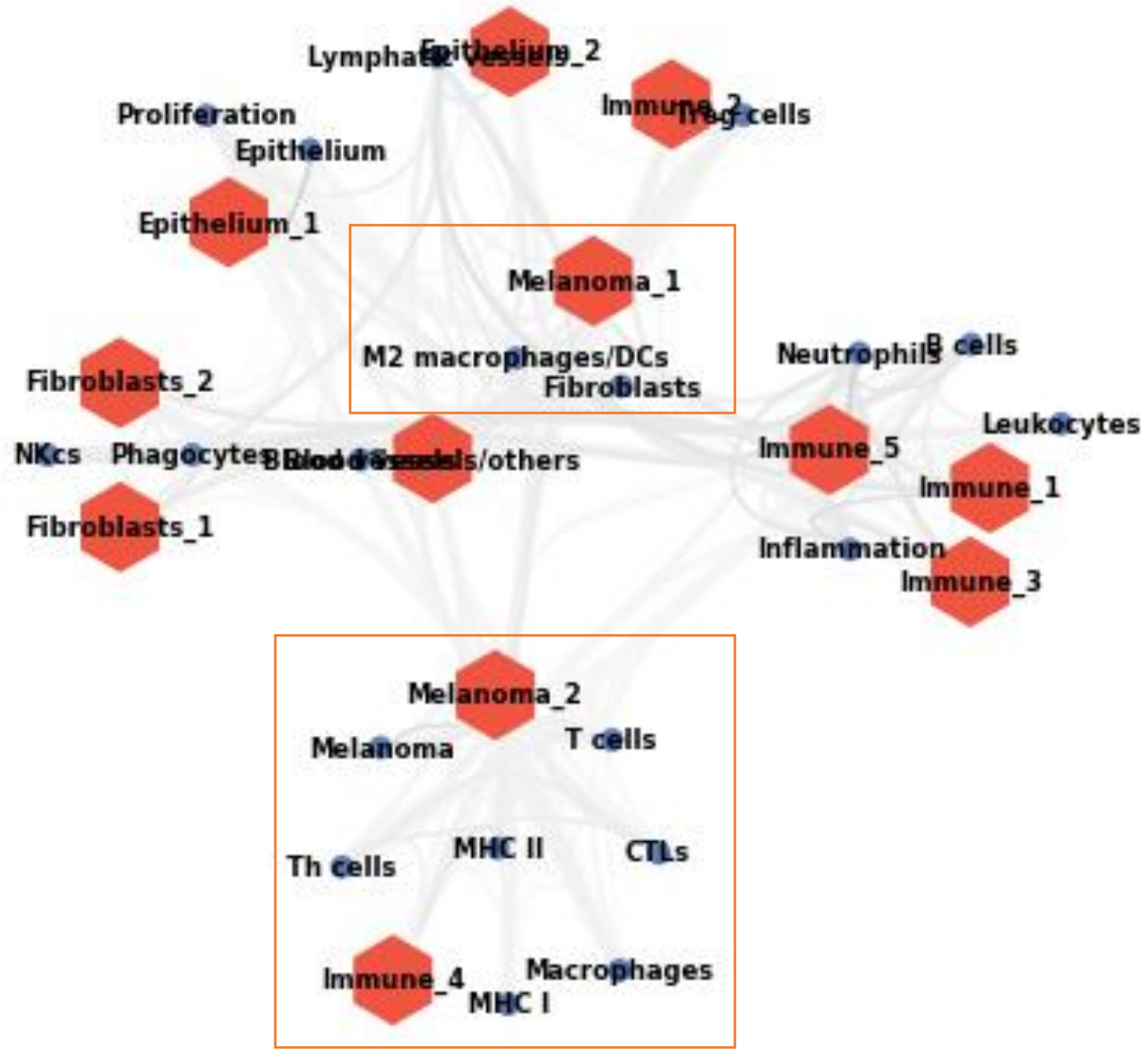




Co-localization between cell types



Cell-type / Niche network

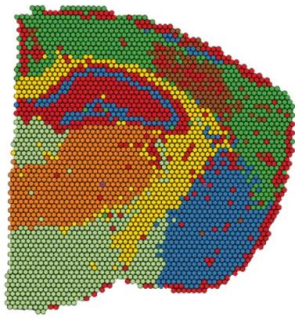


Practical 4: Multiomics Integration - gsMAP

- Prakrithi Pavithra

TensorFlow PyTorch

Python packages



10X Visium

1. Prepare Input Data

ST data

The input ST data must be an h5ad file containing at least the gene expression matrix and spatial coordinates. The gene expression matrix should be stored in the 'layers' attribute, and the spatial coordinates should be in the 'obsm' attribute with the key 'spatial'. Optionally, the h5ad file may include spot (cell) annotations in the 'obs' attribute.

```
AnnData object with n_obs × n_vars = 1270 × 17943
    obs: 'in_tissue', 'array_row', 'array_col', 'imagecol', 'imagerow', 'cell_type'
    var: 'gene_ids', 'feature_types', 'genome'
    uns: 'cell_type', 'spatial'
    obsm: 'spatial'
    layers: 'count'
```

GWAS data

The input GWAS data is a text file containing at least the columns for SNP (rs number), Z (Z-statistics), and N (sample size). Column headers are keywords used by gsMap.

```
zcat gsMap_example_data/GWAS/IQ_NG_2018.sumstats.gz | head -n 5

SNP  A1 A2 Z N
rs12184267 T C 0.916 225955
rs12184277 G A 0.656 226215
rs12184279 A C 1.050 226224
rs116801199 T G 0.300 226626
```

Formatting summary stats compatible with gsMap

The **Z-score** is usually derived from the GWAS **p-value** and the **effect direction** (i.e., the sign of the effect size, β or log(OR)).
The formula is:

$$Z = \Phi^{-1}\left(1 - \frac{p}{2}\right) \times \text{sign}(\beta)$$

where:
p= two-sided GWAS p-value
Φ^{-1}= inverse cumulative distribution function (inverse normal CDF)
sign(β)= +1 if the effect size is positive, −1 if negative

Example
Suppose your GWAS summary statistics contain:

SNP	β	P
rs1	0.25	2×10^{-8}

Then, $\Phi^{-1}(1 - 10^{-8}) \approx 5.61$

Since β is positive, $Z = +5.61$ If β were −0.25, $Z = -5.61$

Convert the summary statistics to the required format.

```
gsmap format_sumstats \
--sumstats 'GIANT_HEIGHT_YENGO_2022_GWAS_SUMMARY_STATS_ALL' \
--out 'HEIGHT'
```

2. Running gsMap in Quick mode

Need to change path according to winter school fol

Copy and paste the following script on terminal to run gsMap

```
gsmap quick_mode \
--workdir '/scratch/project/stseq/Prakrithi/skin_atlas/GWAS/gsmap_test' \
 --sample_name 'melanoma_gsmap' \
--gsMap_resource_dir '/scratch/project/stseq/Prakrithi/skin_atlas/GWAS/gsMap_resource' \
--hdf5_path 'mel48974_raw.h5ad' \
--annotation 'cell_type' \
--data_layer 'count' \
--sumstats_file 'mel.sumstats.gz' \
 --trait_name 'mel'
```

If running for multiple traits on same sample, you can specify a config file with paths to each summary stats file

The `gwas_config.yaml` file includes the following:

```
Height: gsMap_example_data/GWAS/GIANT_EUR_Height_2022_Nature.sumstats.gz
IQ: gsMap_example_data/GWAS/IQ_NG_2018.sumstats.gz
SCZ: gsMap_example_data/GWAS/PGC3_SCZ_wave3_public_INFO80.sumstats.gz
```

The **gsMap** pipeline consists of six primary steps

Step 1: Find Latent Representations The goal is to learn embeddings for every "spot"

Step 2: Generate Gene Specificity Scores (GSS) The GSS is then calculated for each spot in a microdomain by determining the relative rank of each gene's expression within its microdomain compared to its rank across the entire ST section.

Step 3: Generate LD Score In this step, the GSS values are assigned to Single Nucleotide Polymorphisms (SNPs).

Step 4: Spatial LDSC This step performs the core association analysis using **stratified LD score regression (S-LDSC)**. It assesses whether heritability for a specific trait is disproportionately enriched in SNPs that have higher GSS for a given spot, resulting in an enrichment *P* value for every spot.

Step 5: Cauchy Combination (Optional) To evaluate the significance of entire spatial regions or cell types rather than just individual spots, gsMap uses the **Cauchy combination test**

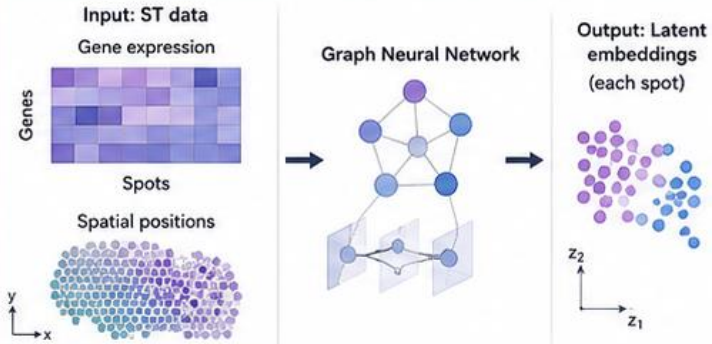
Step 6: Report Generation (Optional) The final step creates a comprehensive report featuring **visualizations of the mapping results** and diagnostic plots. This includes highlighting top trait-associated genes and plotting heritability enrichment across the tissue sections.

These files generated can be used to customize the plots for further downstream analysis

gsMap Pipeline: From GWAS Summary Statistics to Spatially Resolved Trait Associations

1. Find Latent Representations

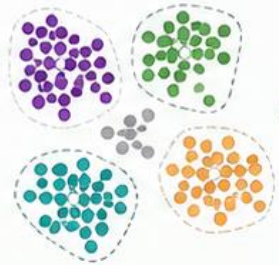
Learn embeddings for every spot by integrating gene expression with spatial positions using a GNN.



The latent space reduces technical noise and sparsity.

2. Generate Gene Specificity Scores (GSS)

Identify homogeneous spots (microdomains) in latent space and compute gene specificity within each microdomain.



Gene Specificity Score (GSS)

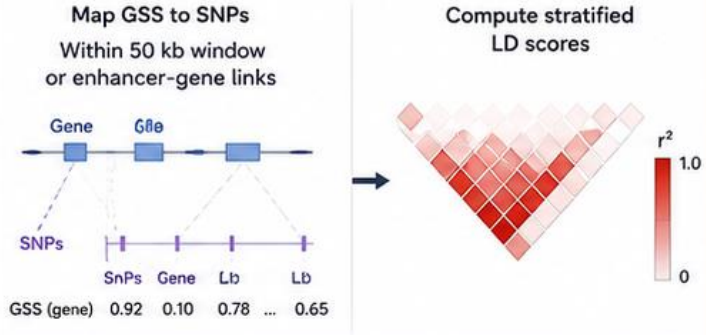
For each spot and gene:
Rank in microdomain
vs. rank in whole section

Gene	Rank in Microdomain	Rank in Section	GSS
Gene 1	12	432	0.97
Gene 2	5	120	0.92
...	...	...	...
Gene n	30	800	0.88

GSS ranges from 0 to 1.
Higher score = more specific to the microdomain.

3. Generate LD Score

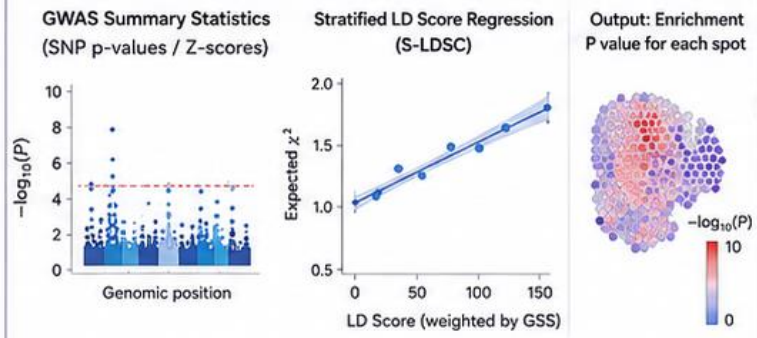
Assign GSS to SNPs using genomic annotations, then compute stratified LD scores.



Result: LD score for each SNP weighted by GSS.

4. Spatial LDSC

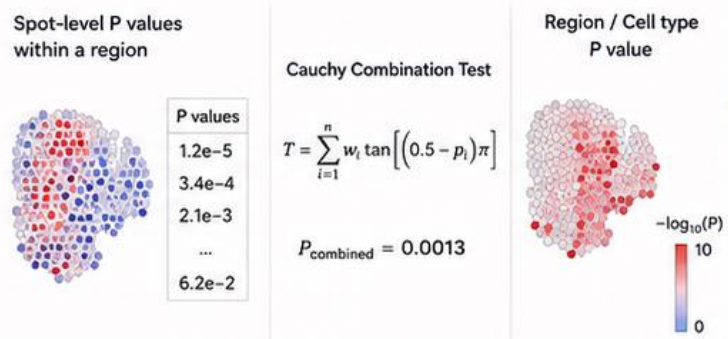
Use stratified LD score regression to test whether heritability is enriched in SNPs with high GSS for each spot.



Identifies spots where trait heritability is significantly enriched.

5. Cauchy Combination (Optional)

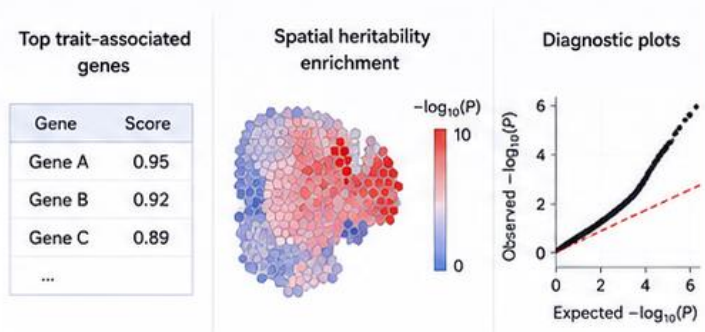
Aggregate P values of spots within a region or cell type to obtain a robust region-level association.



Provides significance for spatial regions or cell types.

6. Report Generation (Optional)

Generate comprehensive reports with visualizations and diagnostic plots.



Deliver interpretable results and quality control diagnostics.

ST: Spatial Transcriptomics **GNN:** Graph Neural Network **GSS:** Gene Specificity Score **LD:** Linkage Disequilibrium **LDSC:** LD Score Regression

3. Running gsMap in Quick mode

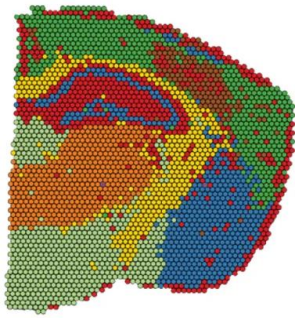
- The final reports are generated in melanoma_gsmap/report/mel
- Browse through the diagnostic info and HTML report

Practical 5: Machine Learning

- Xiao Tan

TensorFlow PyTorch

Python packages



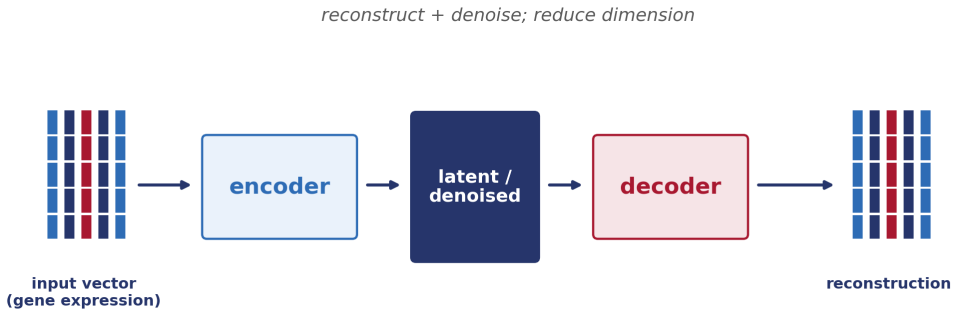
10X Visium

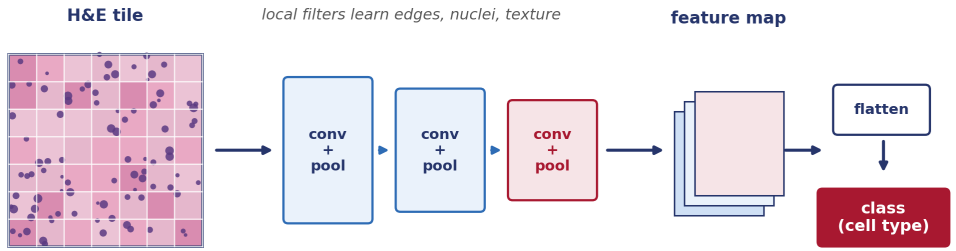
Deep Learning 01: Autoencoder + CNN

Practical 005 · Notebook 01

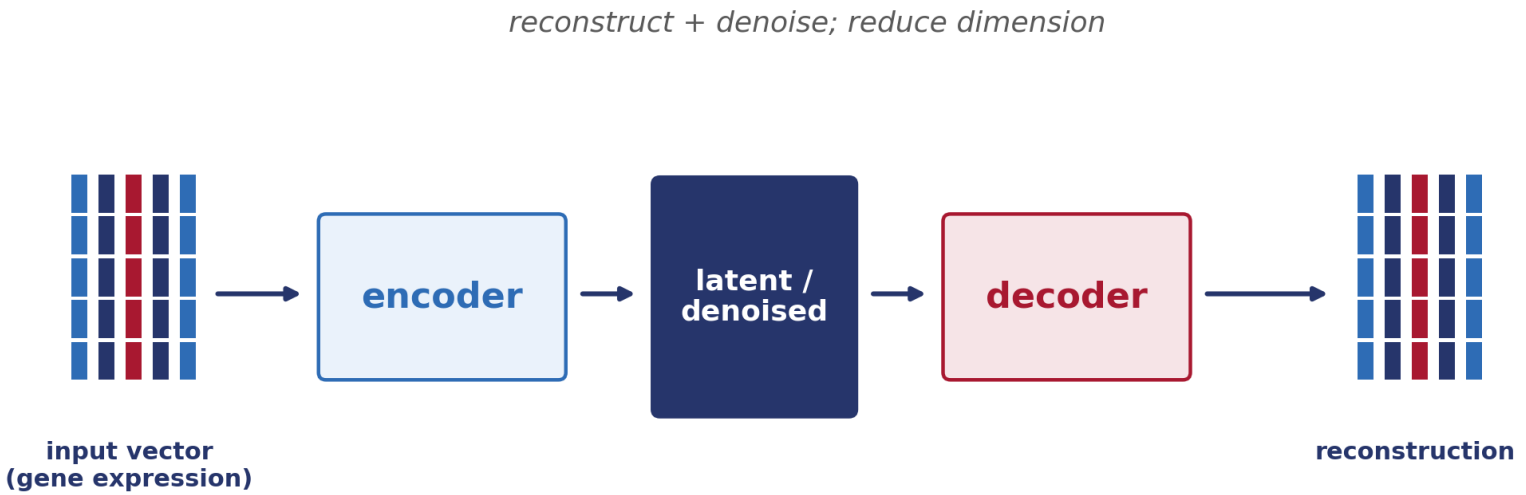
What we'll do in this session

* First: cluster Visium expression so we have a spatial structure to preserve.
* Second: train an autoencoder to reconstruct and denoise gene expression.
* Third: train a small CNN that maps H&E tiles to cell-type labels.



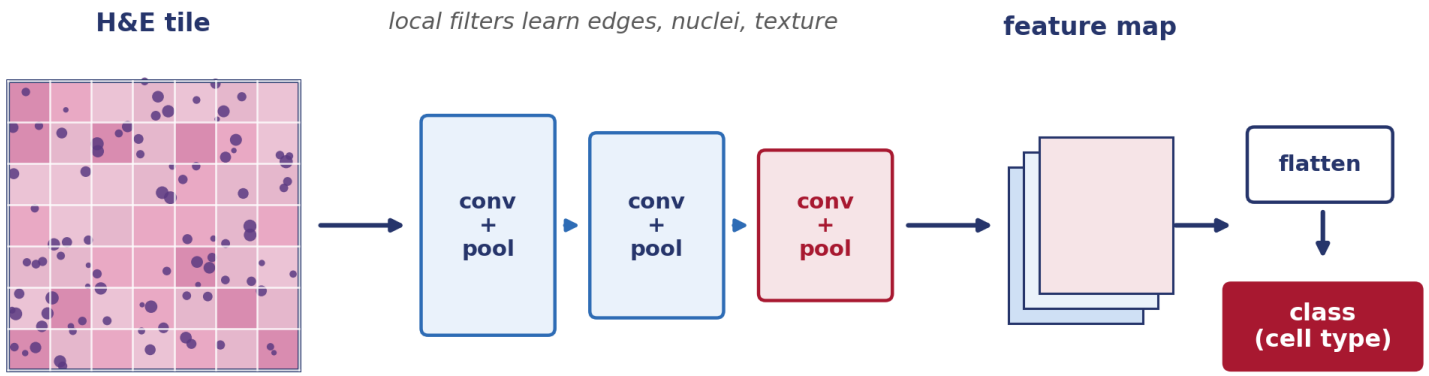


Autoencoders compress, denoise, and reconstruct expression



- The encoder maps high-dimensional counts into a compact latent representation.
- The decoder is trained to reconstruct the input, forcing the latent space to keep signal.
- For spatial omics, that latent space can denoise expression and support downstream clustering.

CNN concept: pixels to morphology features



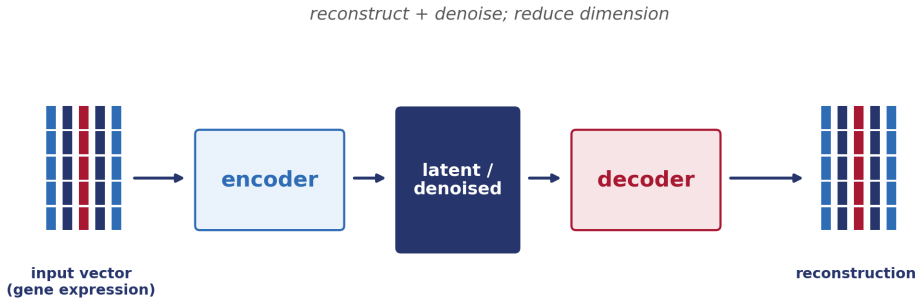
* A CNN stacks local filters to turn pixels into increasingly abstract image features.
* Here, each training item is a cropped H&E tile plus an expression-derived label.
* This turns morphology into a supervised image-classification task.

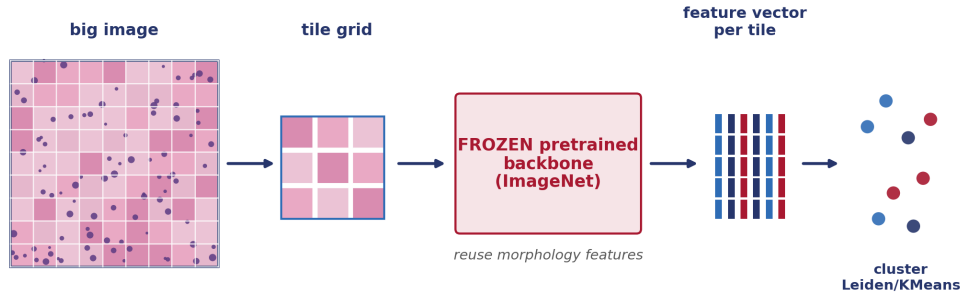
Deep Learning 02: Autoencoder + Transfer Learning

Practical 005 · Notebook 02

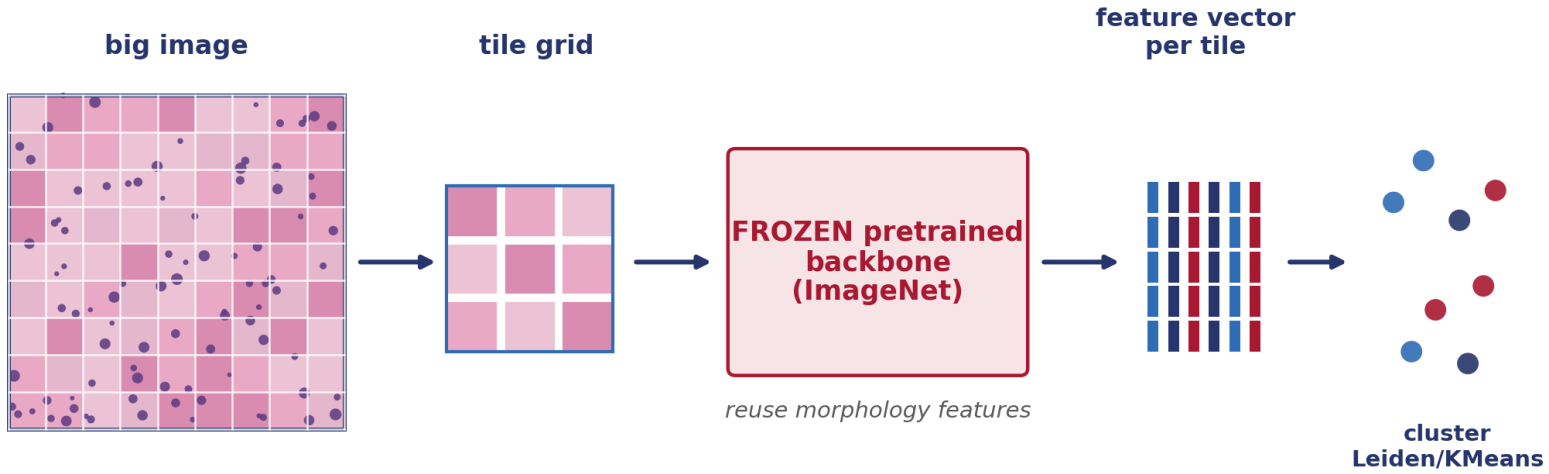
Overview and practical steps

* Load the 10x breast-cancer Visium sample and inspect basic QC metrics.
* Train an expression autoencoder, then examine total and per-spot loss.
* Use a pretrained ResNet backbone as an image-feature extractor.
* Cluster morphology features and integrate them with the expression latent space.





Transfer-learning concept: reuse a pretrained backbone



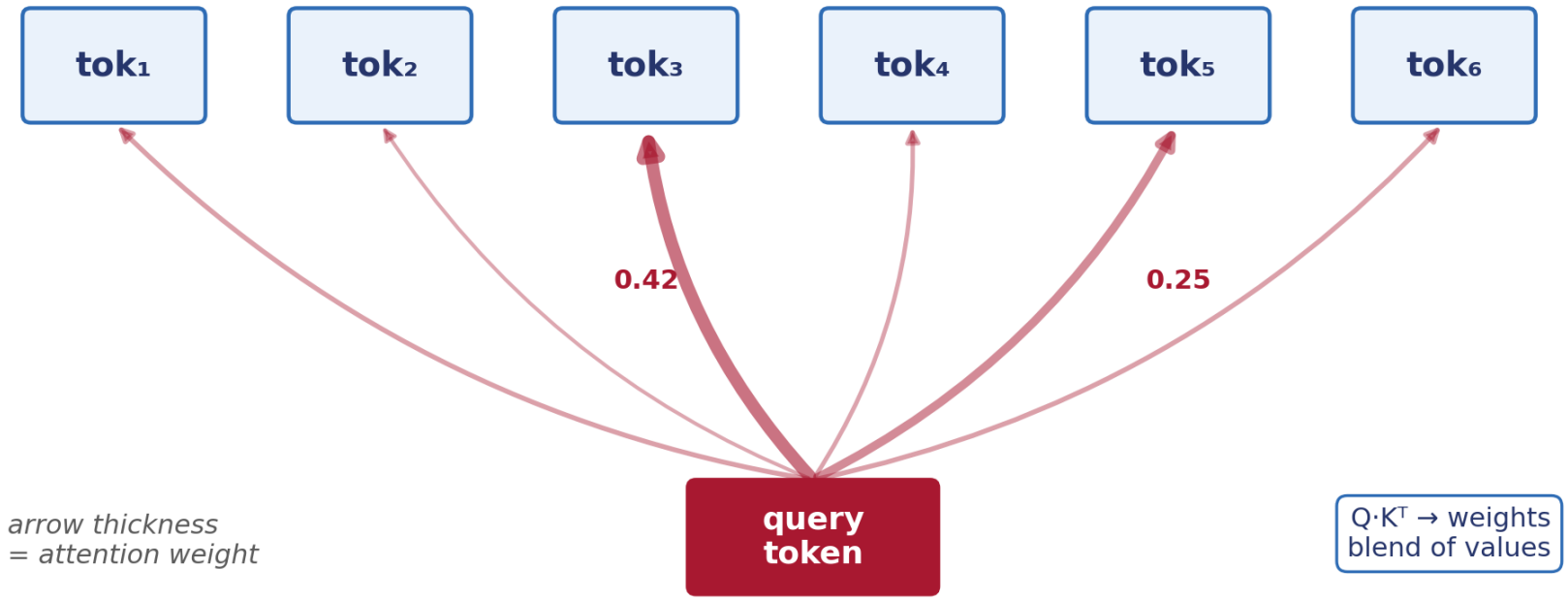
* Instead of training a full CNN, pass each tile through a frozen ImageNet-pretrained model.
* The model activations become reusable morphology features.
* This is the practical predecessor to modern histology foundation models.

Deep Learning 03: Transformer Genes

Module 004 · Notebook 03

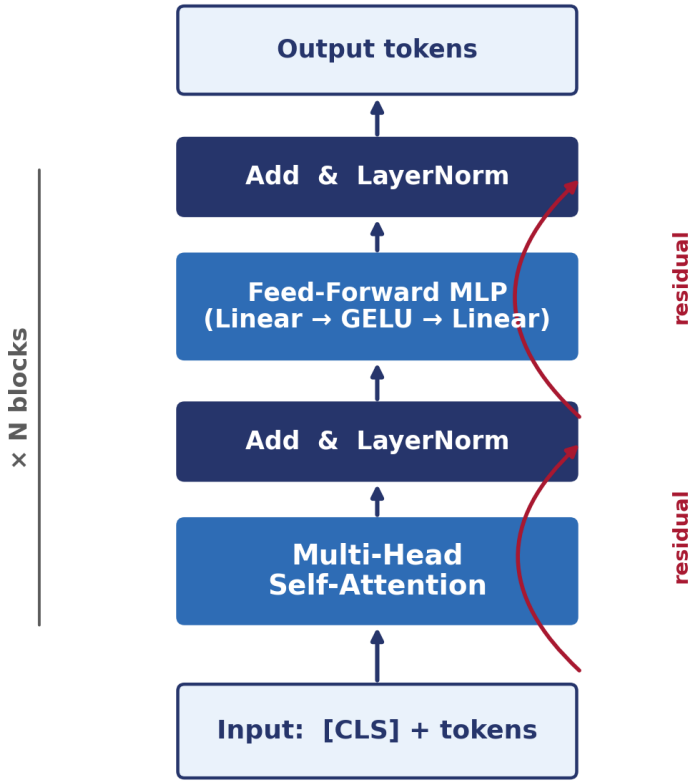
From CNNs and autoencoders to the Transformer

Attention lets every token mix information from all others



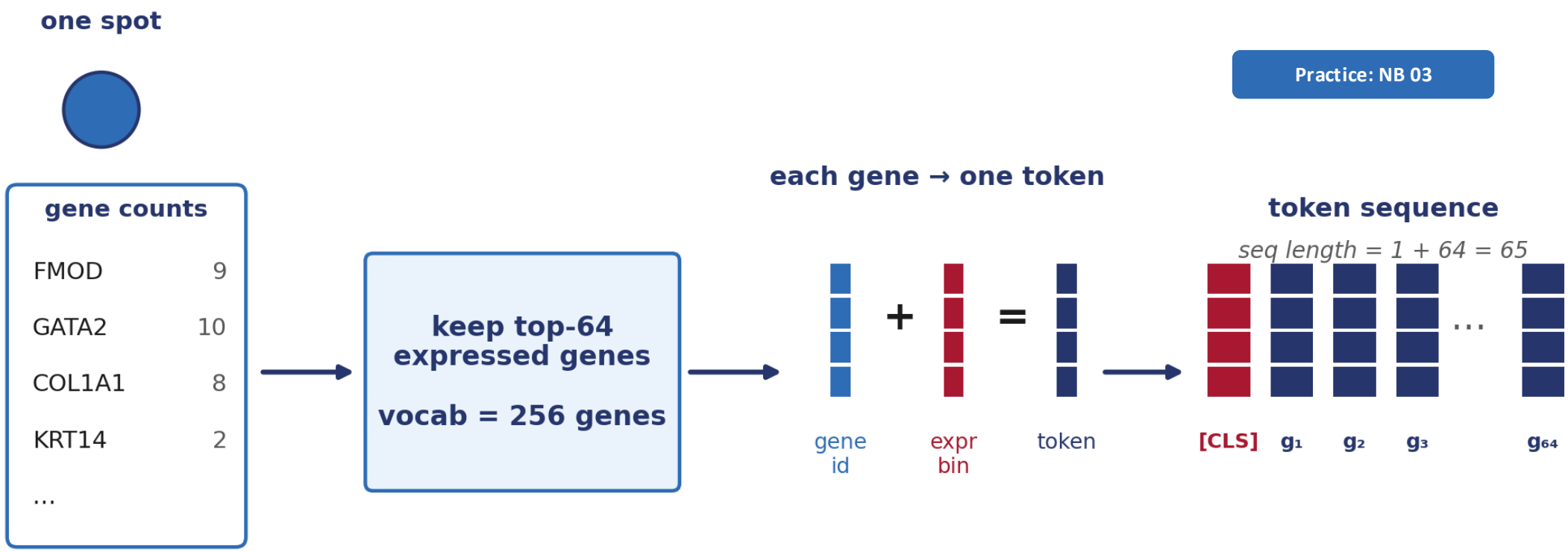
- A “token” is one unit of input — can be a gene or image patch.
- Each token forms a query and reads every token through learned weights.
- No convolution, no recurrence: attention is the only mixing step.

A Transformer block: attention + MLP, wrapped in residuals



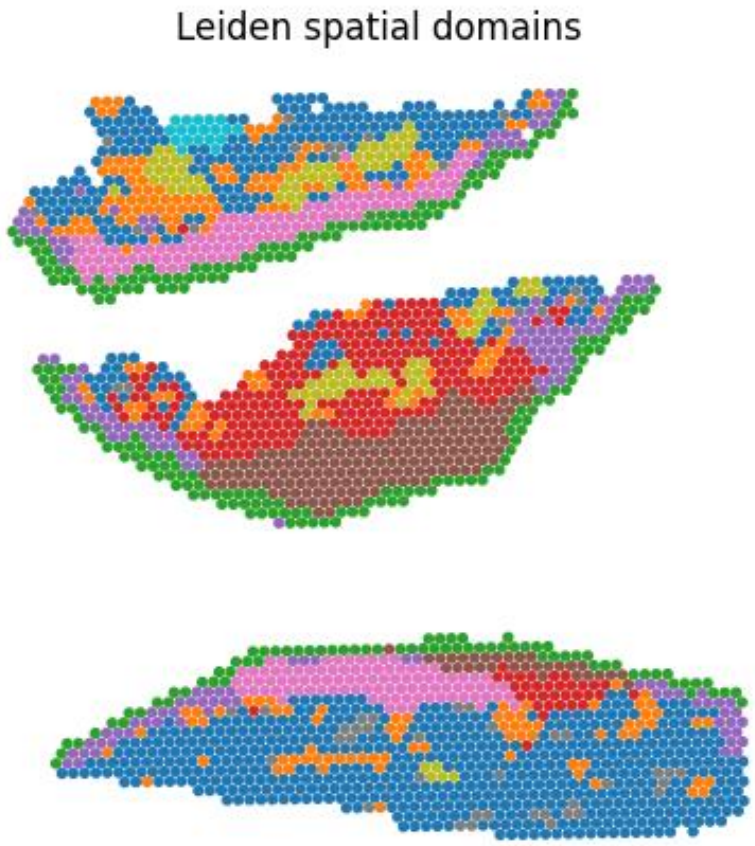
- Multi-head self-attention: tokens exchange information.
- Feed-forward MLP: transform each token on its own.
- Residual + LayerNorm around each sub-layer → stable training.
- Stack N blocks; a special [CLS] token pools a whole-sequence summary.

From gene counts to tokens the Transformer can read



- Vocabulary = 256 most-variable genes; keep the top 64 expressed per spot.
- Each token = gene-identity embedding + expression-bin embedding (10 deciles).
- Prepend a [CLS] token → sequence length 65.

A small from-scratch Transformer classifies spatial domains

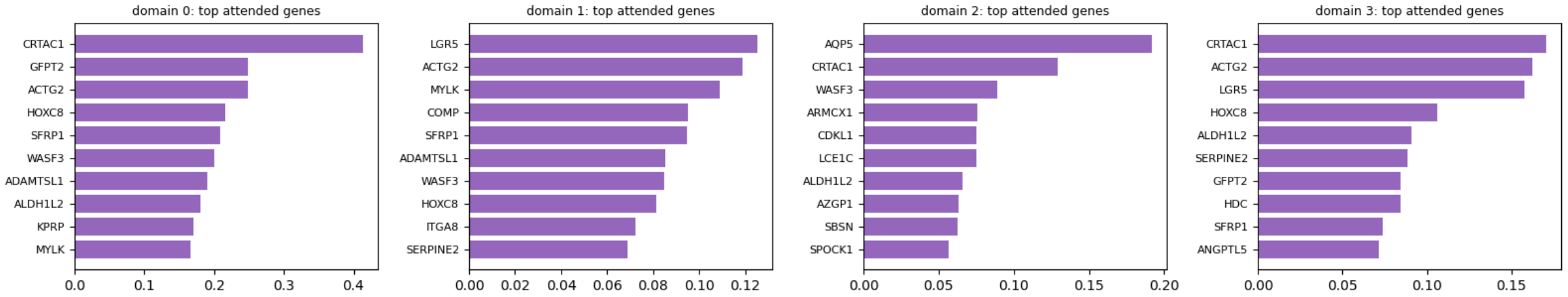


> ***"A spot is a sentence, and the genes it expresses are the words."***

- Foundation models scGPT and Geneformer treat each cell/spot as a sequence of gene tokens.
- Self-attention learns which genes co-vary — data-driven gene programs, not fixed pathways.
- Our goal: classify spatial domains, then read the attention to see which genes drove it.

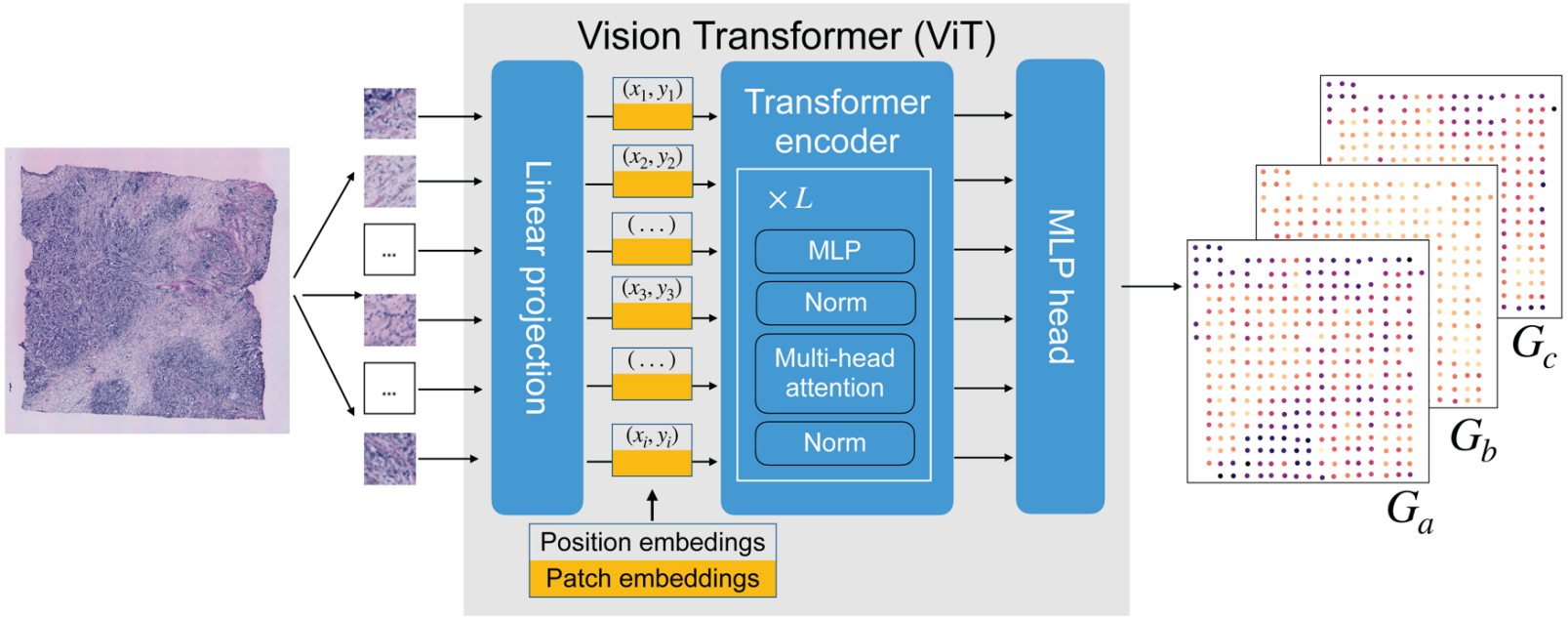
- Labels: Leiden clustering gives 10 spatial domains
- GeneTransformer: embeddings → 2 blocks → classify from the [CLS] token.
- ~85,000 parameters; trains on CPU during the workshop.

Reading attention reveals each domain's marker genes



- Average the [CLS] → gene attention over all spots of a domain.
- The top-attended genes are recognisable markers — the model is interpretable.
- Attention is a built-in explanation, not a post-hoc saliency method.

Same Transformer, new modality: the H&E image

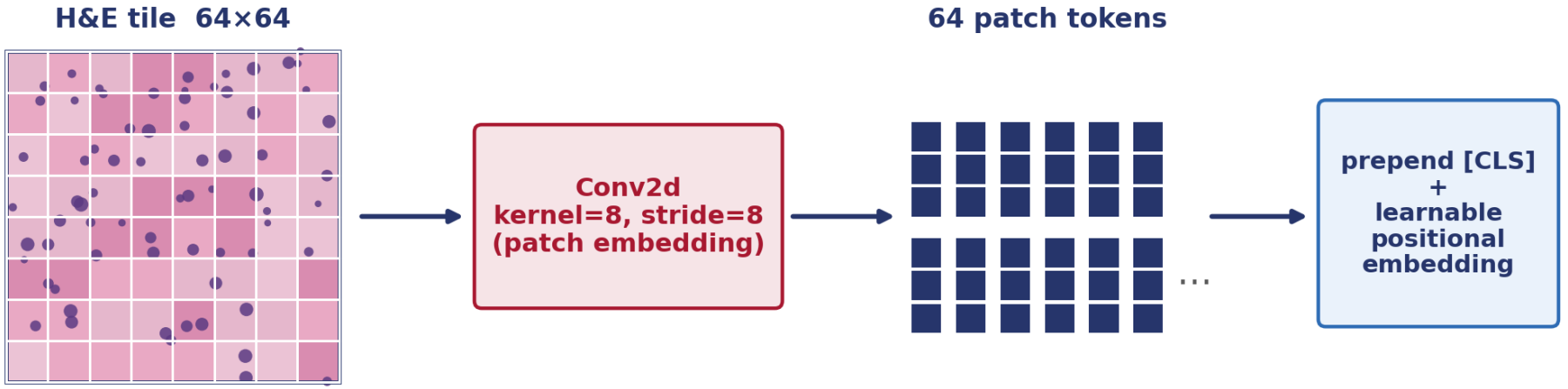


- An image tile is a grid of patches; each patch becomes a token (ViT, Dosovitskiy 2020).
- Task — predict expression from morphology.
- The TransformerBlock is identical to NB 03 — only the input changes.

Patch embedding turns a tile into 64 patch tokens

Practice: NB 04

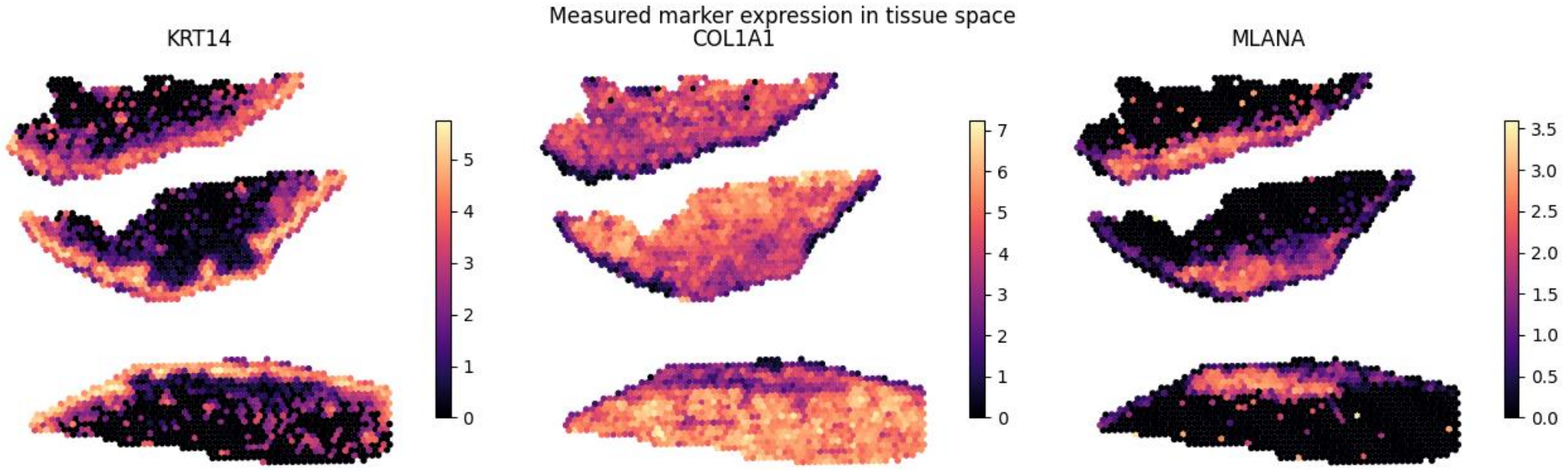
the only new piece vs. the gene Transformer



split into 8×8 = 64 patches

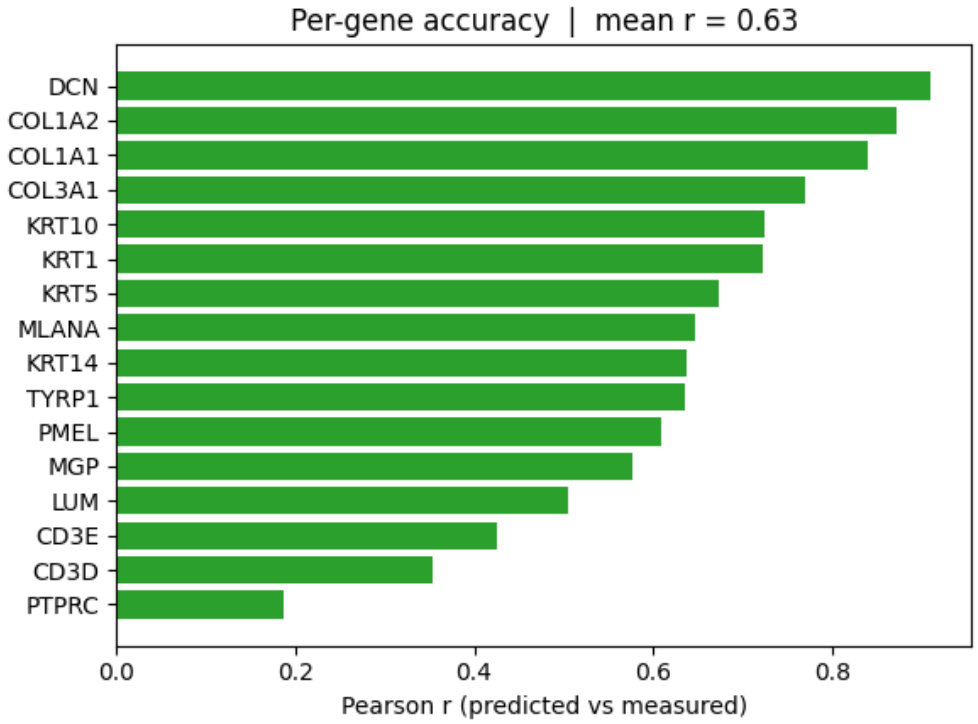
- A strided Conv2d (kernel = stride = 8) slices the 64×64 tile into 8×8 = 64 patches.
- Prepend [CLS]; add a learnable positional embedding.
- From here it is the same Transformer as the gene notebook.

Predict 16 interpretable marker genes from each tile



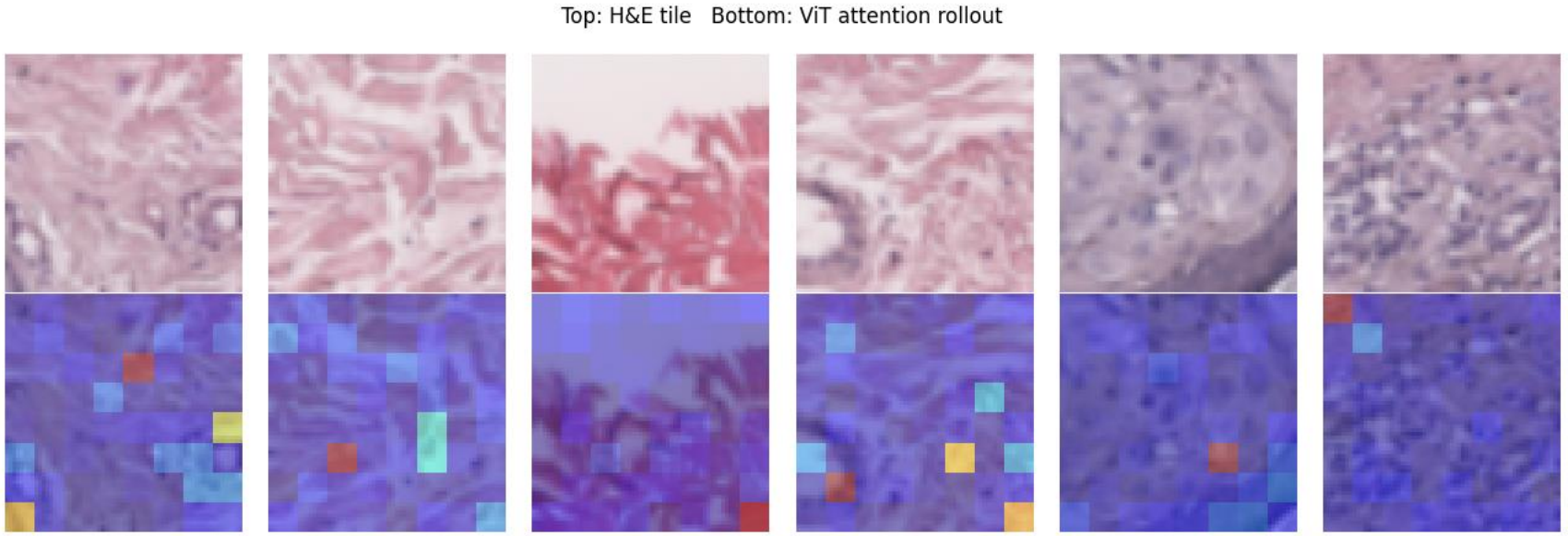
- Targets: keratins (epidermis), collagens (dermis), melanocyte and immune markers — 16 genes.
- Expression is log-normalised and z-scored per gene; loss = mean-squared error.
- Input: a 64×64 H&E tile cropped under each Visium spot.

Structural genes are predictable from morphology



- Mean Pearson r ≈ 0.63 across the 16 genes on validation spots.
- Collagens and keratins score highest — they have a distinctive tissue appearance.

Attention rollout shows where the ViT looks



- Rollout multiplies each layer’s attention (with the residual) back to the input patches.
- The [CLS] token concentrates on cellular / structural regions of the tile.
- Same interpretability idea as NB 03 — now over image space instead of genes.

One attention machine, two views of spatial data

NB 03 · genes as tokens

- Spot → sequence of gene tokens
- Supervised domains + attention read-out
- Self-supervised = mini scGPT / Geneformer

NB 04 · patches as tokens

- Tile → sequence of patch tokens
- Regress expression from morphology
- Rollout shows where it looks

- Same TransformerBlock, same [CLS] read-out — only the tokens differ.
- Attention gives interpretability in both settings.
- Bridges to real models: scGPT / Geneformer (genes) and ViT / HisToGene / Hist2ST (histology).