CS502: Deep Learning in Biomedicine
Single-cell Genomics & Transformers

Maria Brbić

Fall 2023

École
polytechnique
fédérale
de Lausanne

Last Week: Recap

- What we covered last time: Transformers

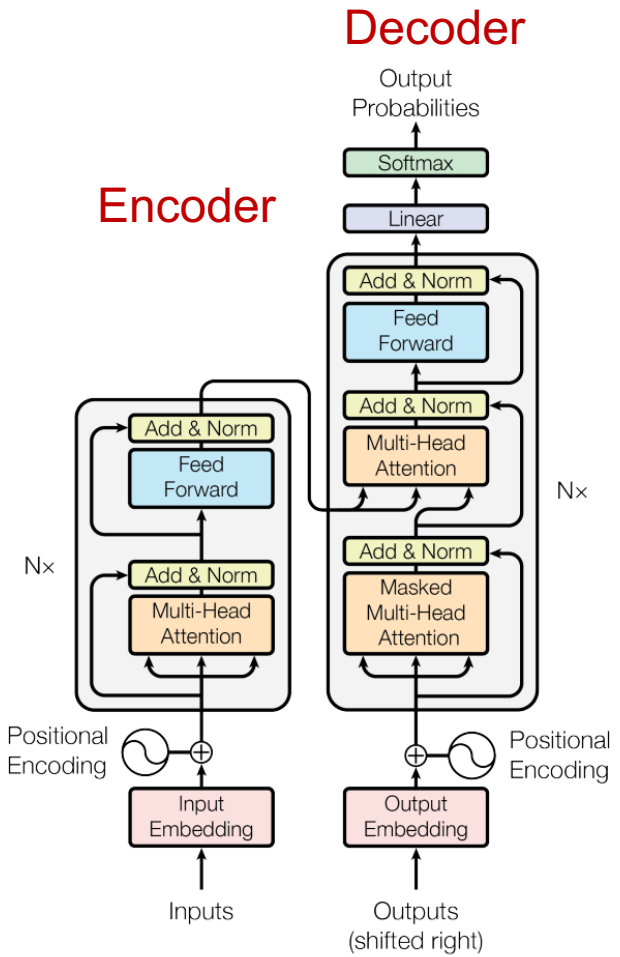


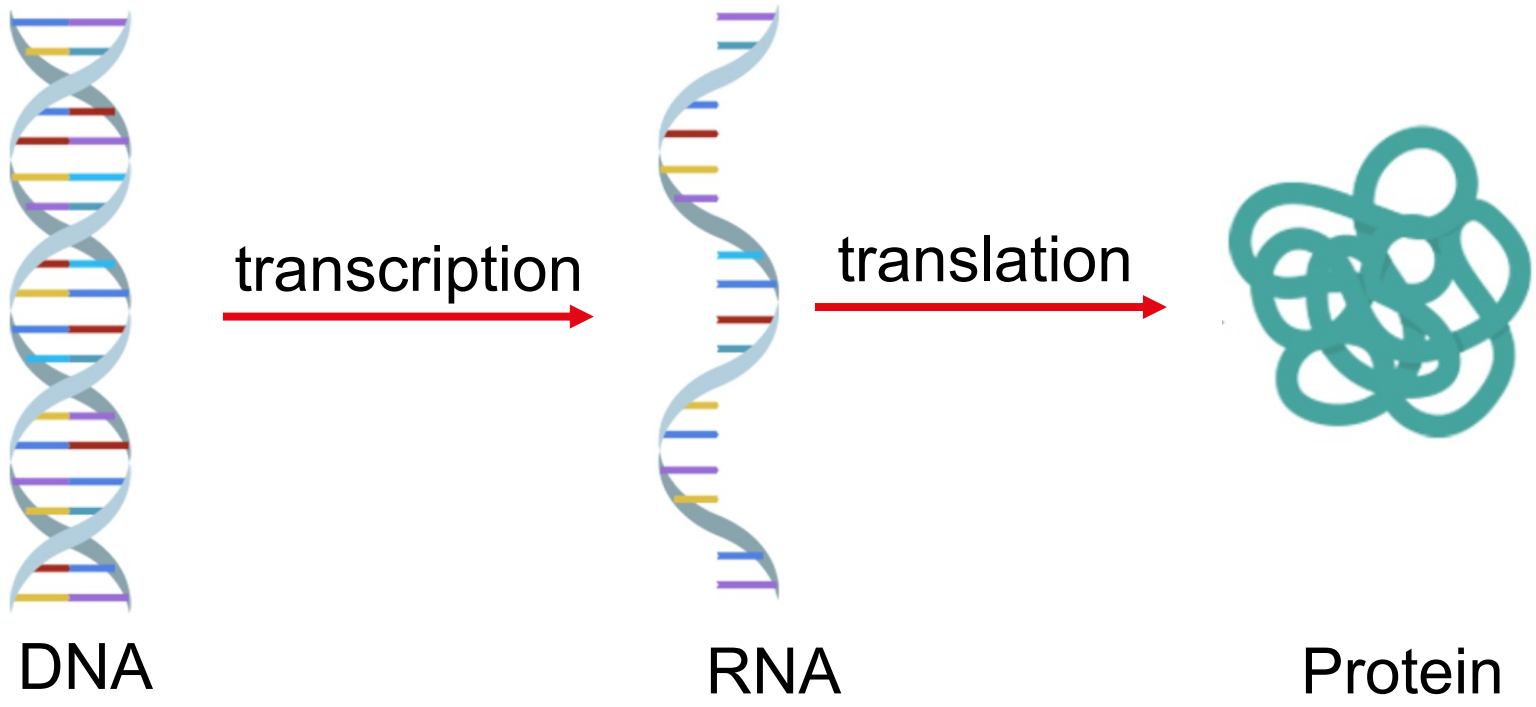
Figure from Vaswani et al. *NeurIPS* 2017

- Transformers
  - Process sequence at once instead of sequentially!

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Biomedical Applications: Biological Sequence Modeling

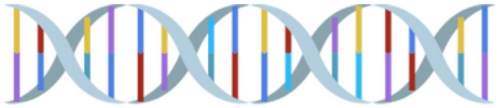
Information Flow in Molecular Biology



- DNA gets transcribed to RNA
- RNA gets translated to protein

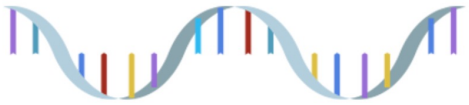
Information Flow in Molecular Biology

Sequences!

DNA 

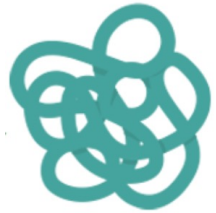
... CACGTAGACTGAGGACTCCTCTTC ...

transcription

RNA 

... GUGCAUCUGACUCCUGAGGAGAAG ...

translation

Protein 

... V H L T P E E K ...

Biological Sequence Analysis

- Biological sequence analysis is one of the fundamental applications of computational methods in molecular biology
  - DNA, RNA and protein sequence analysis
- Traditional techniques:
  - K-mers (Koonin and Galperin, 2003)
  - CNNs (e.g., Zhou and Troyanskaya, 2015, Kelley et al., 2016)
  - RNNs and LSTMs (e.g., Jurtz et al., 2017)
- Nowadays:
  - **Transformers**

An Overview of Transformer Related Works

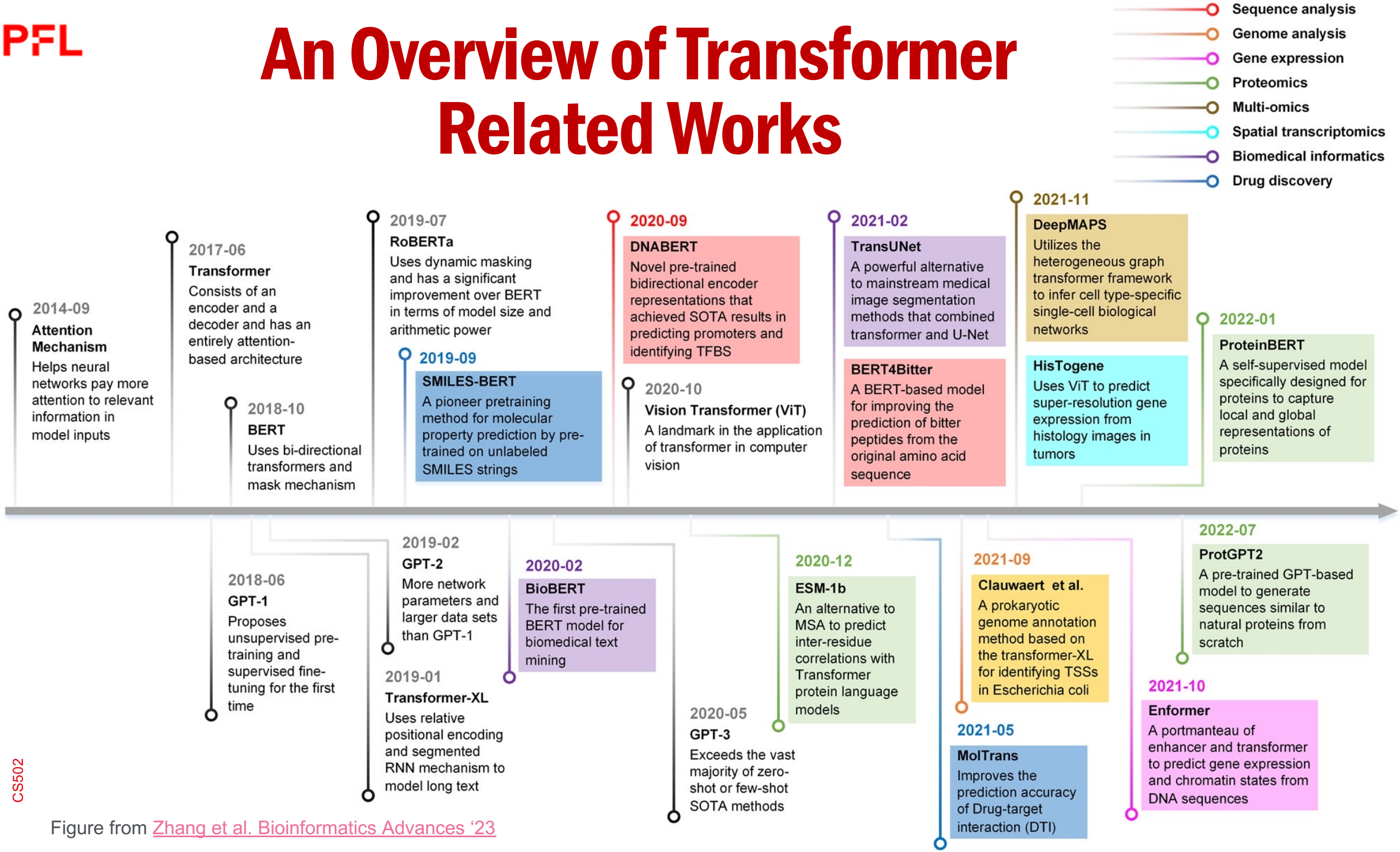
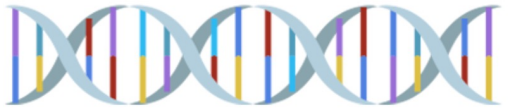


Figure from Zhang et al. Bioinformatics Advances '23

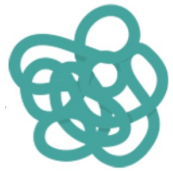
Transformer-Based Models for Biological Sequences: Examples

- **DNA sequences**
  - DNABert (Ji et al., 2021)
  - Enformer (Avsec et al., 2021)
  - HyenaDNA (Nguyen et al., 2023)
  - DNAGPT (Zhang et al., 2023)



- **RNA:**
  - GeneFormer (Theodoris et al., 2023)
  - scGPT (Cui et al., 2023)
  - scFoundation (Hao et al., 2023)

- **Protein sequences:**
  - ESM1b (Rives et al., 2021)
  - ESM2 (Lin et al., 2023)
  - ProGen (Madani et al., 2023)



DNABERT

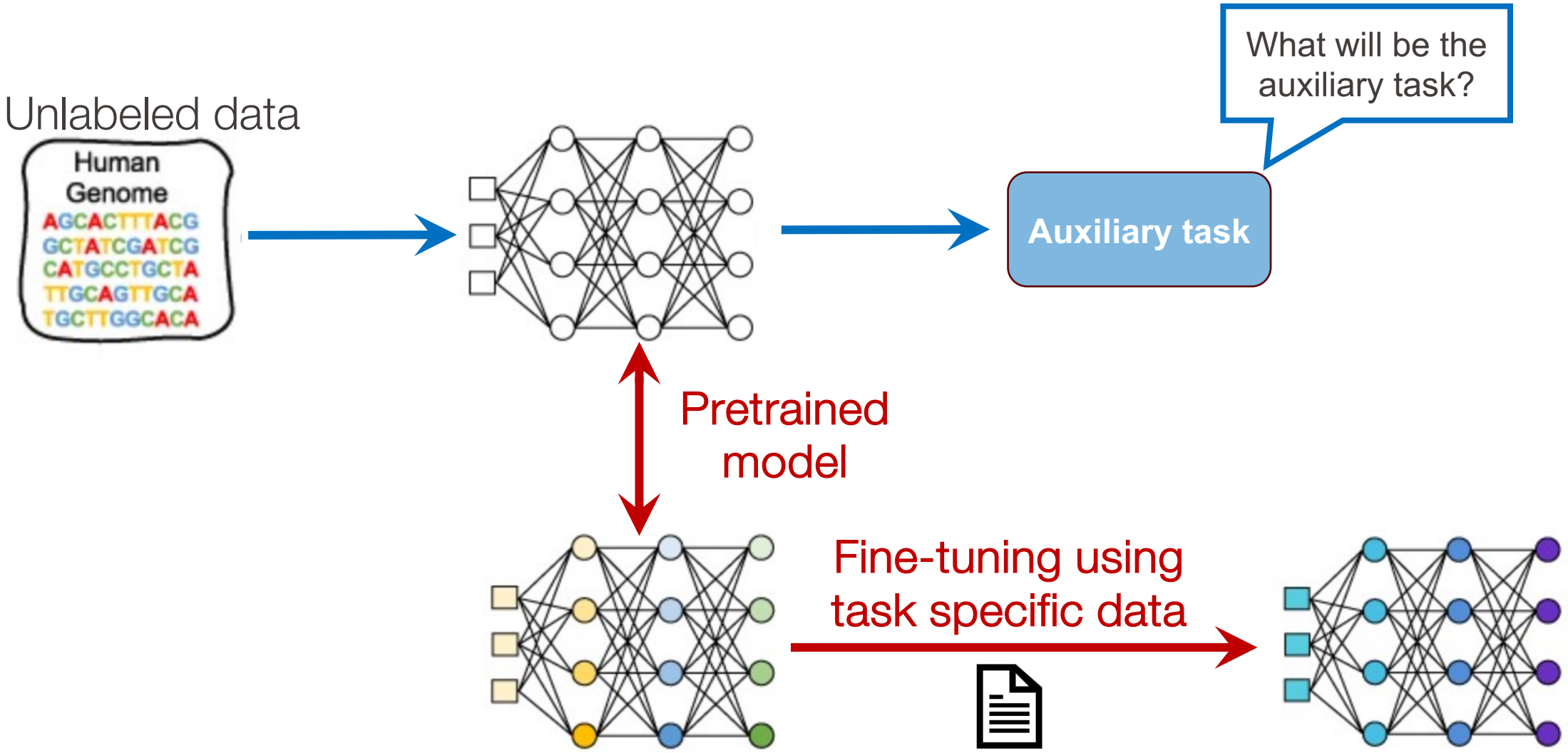
Ji, Zhou, Liu, Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* 2021

DNABERT: Pretrain & Fine-tune

- Pre-trained bidirectional encoder for understanding of genomic DNA sequences
  - Bidirectional: use information from the entire sequence regardless of the position
- Based on masked language model BERT (Devlin et al., 2018)
- Encoder-only transformer

- **Approach:**
  - Pretrain the model using large amount of available data on the auxiliary task → self-supervised learning
  - Fine-tune the model on task-specific data

DNABERT: Pretrain & Fine-tune



DNABERT: Pretraining

Masked token prediction!

- But, let's first construct tokens!
  - In NLP, tokens are words what could be tokens in DNA sequences?

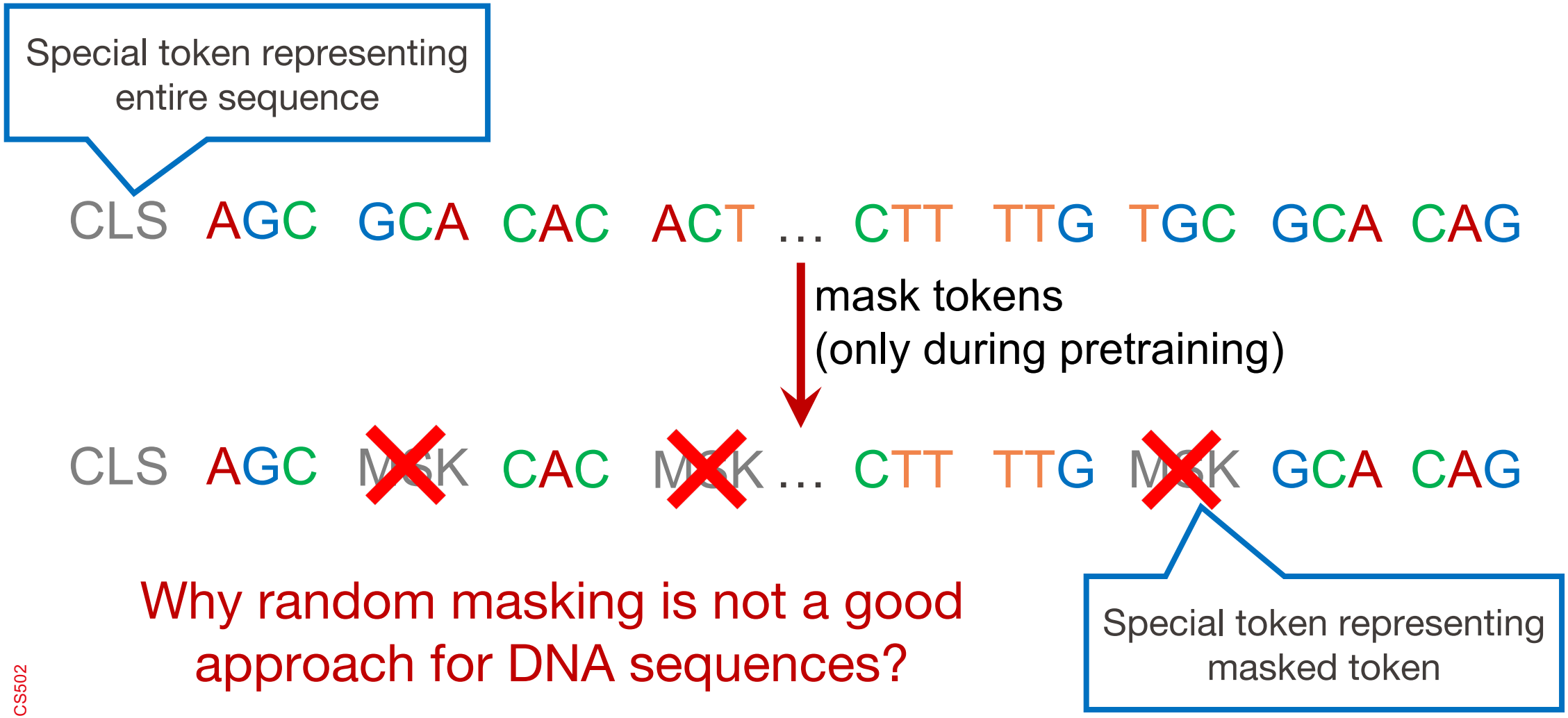
N-grams!

AGCACTGCTATCATGCTTGCAG

tokenize

AGC GCA CAC ACT … CTT TTG TGC GCA CAG

DNABERT: Pretraining



Why random masking is not a good approach for DNA sequences?

DNABERT: Pretraining

Special token representing entire sequence

CLS AGC GCA CAC ACT … CTT TTG TGC GCA CAG

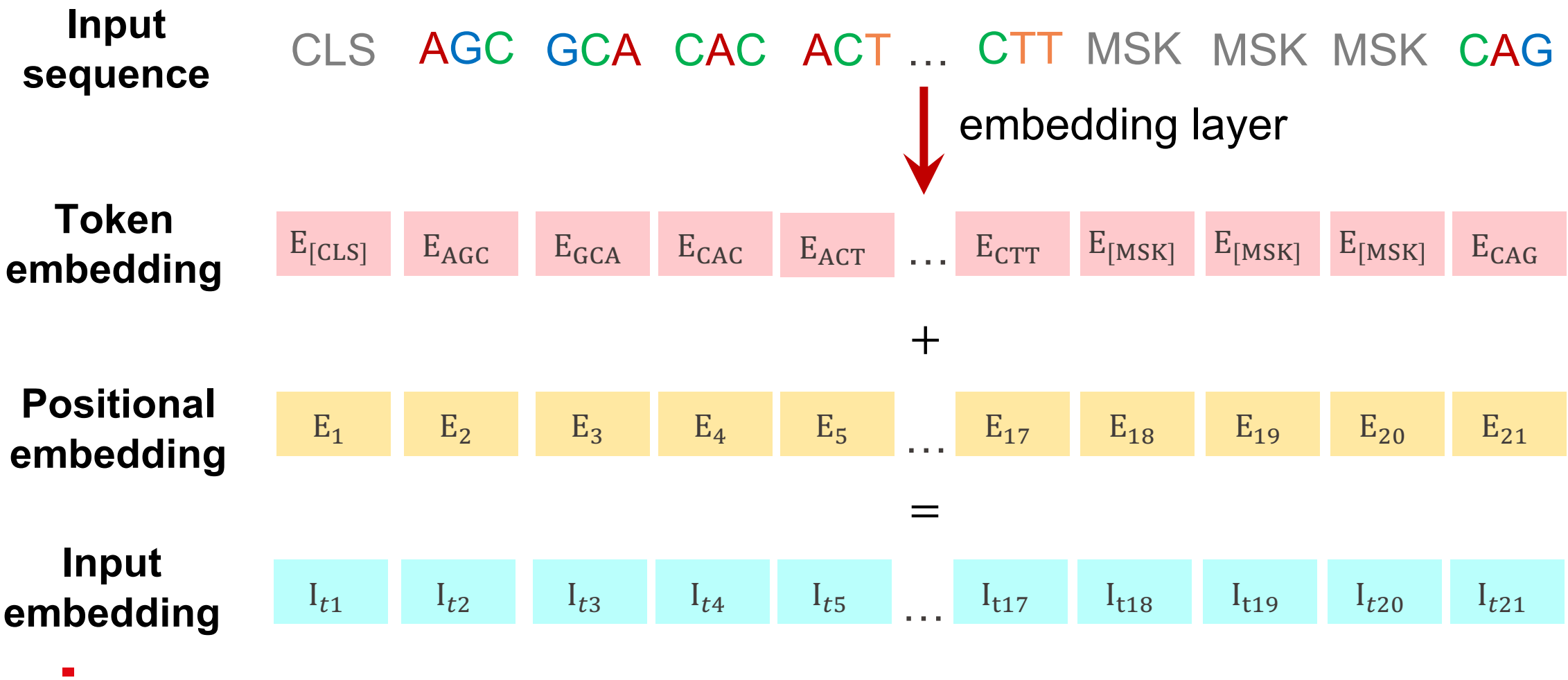
mask tokens
(only during pretraining)

CLS AGC GCA CAC ACT … CTT MSK MSK MSK CAG

Mask contiguous tokens!

- Randomly mask tokens
  - DNABERT masks 15% tokens
  - In DNA, we need to mask contiguous k-length spans of k-mers

DNABERT: Pretraining

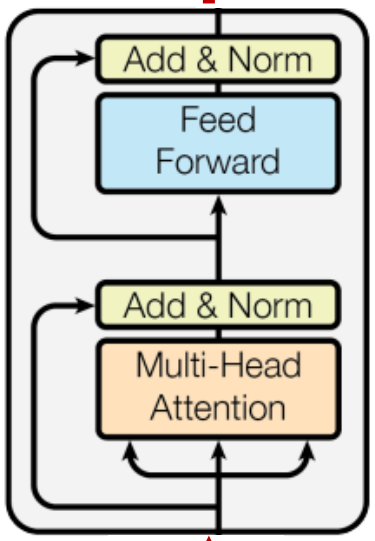


DNABERT: Pretraining

Last hidden state

O_{t1} O_{t2} O_{t3} O_{t4} O_{t5} … O_{t17} O_{t18} O_{t19} O_{t20} O_{t21}

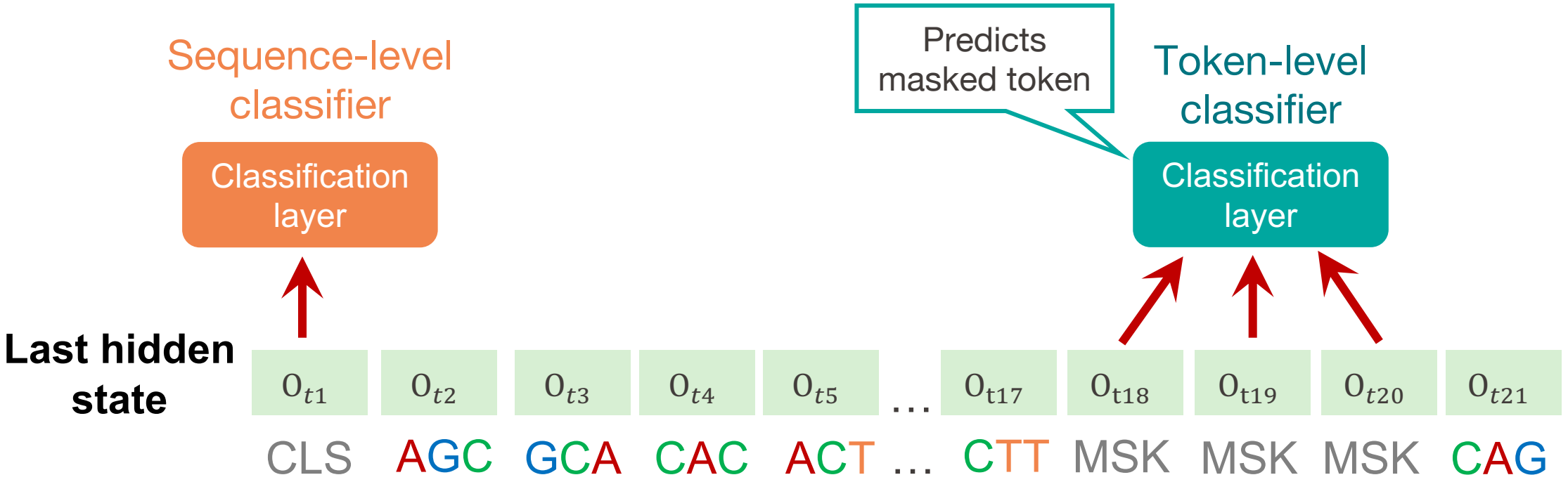
12 ×



Input embedding

I_{t1} I_{t2} I_{t3} I_{t4} I_{t5} … I_{t17} I_{t18} I_{t19} I_{t20} I_{t21}

DNABERT: Pretraining



- Sequence-level classifier could be added but not used in DNABERT
  - Given a pair of sequences, predicts whether one follows after the other one

DNABERT: Pretraining

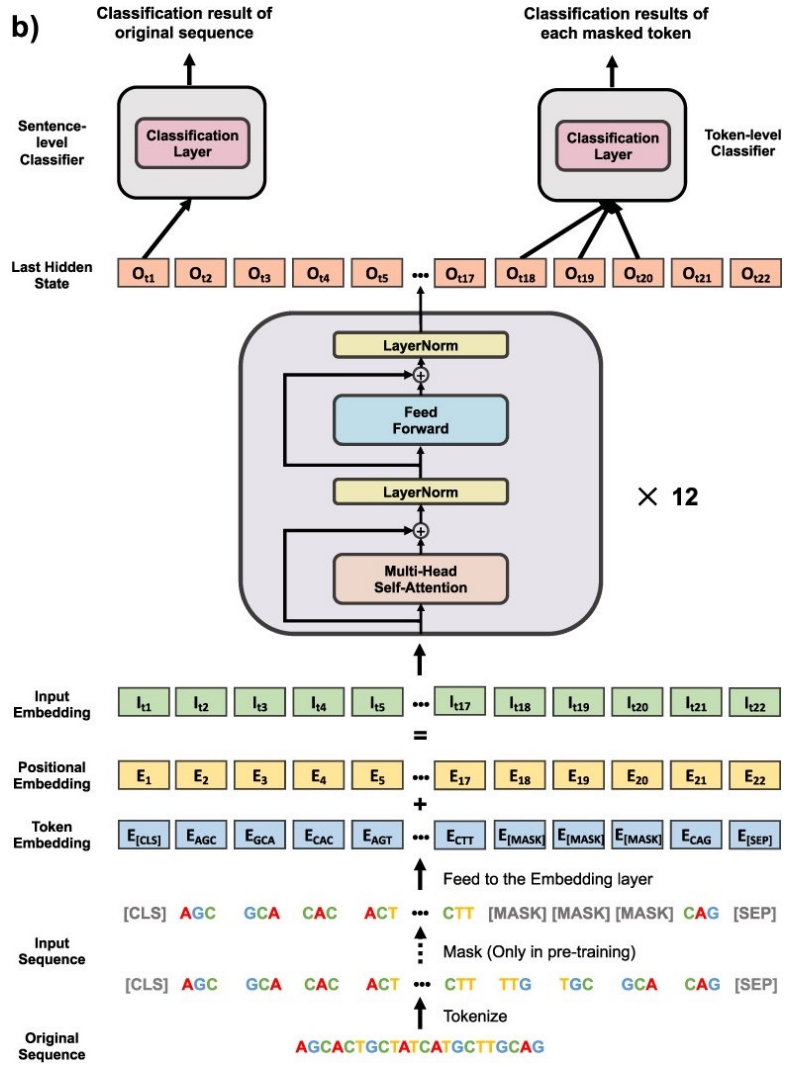
- Dataset: Reference human genome
  - Length of sequence between 5 and 510 nucleotides
- For each sequence, randomly mask regions of k contiguous tokens that constitute 15% of the sequence
- DNABERT predicts the masked sequences based on the remainder tokens
- Cross-entropy loss function: $\mathcal{L} = \sum_{i=0}^{N} -y_i' \log(y_i)$

  y_i': ground truth

  y_i: predicted probability for a class i
- Different values of k for k-mers:
  - Experiments with $k = 3 \ldots 6$

DNABERT: Architecture Recap

- DNABERT architecture is composed of multiple sequential blocks:
  - Input embedding (including positional embedding)
  - Transformer encoder layers (x12)
  - Classifier (one fully connected layer)



DNABERT: Fine-tuning Parameters Optimization

- During DNABERT pretraining:
  - Optimization of all layers
- During DNABERT fine-tuning:
  - Only optimization of the final layer, the classifier (fully connected layer)
  - “Freezing” of the other layers (keep their parameters fixed)

DNABERT: Fine-tuning Applications

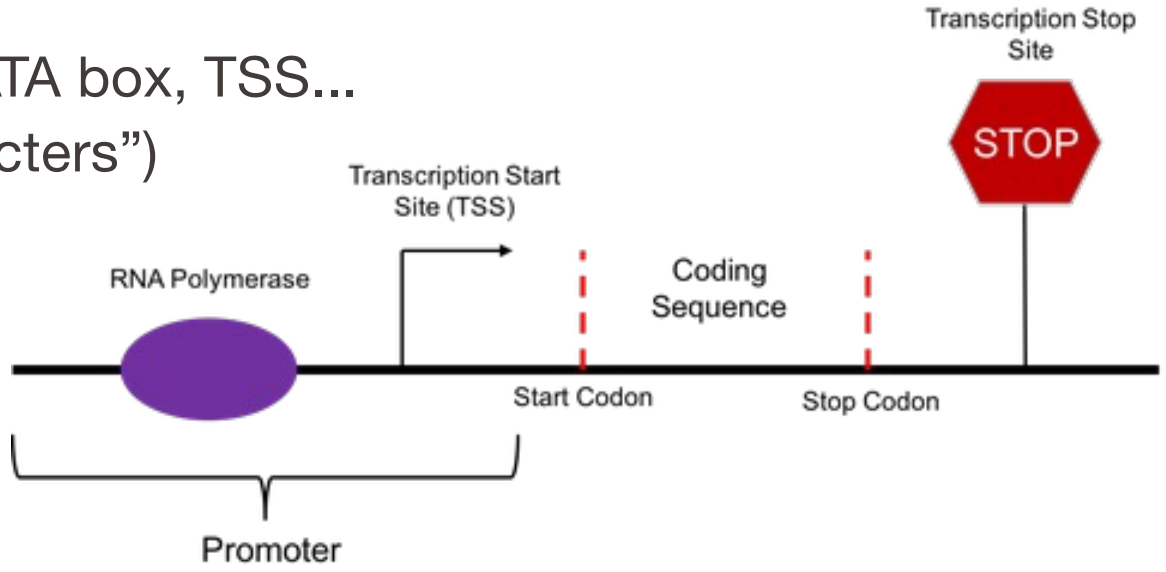
- DNABERT is able to solve different types of tasks:
  - **Token-level tasks**
    - Example: seen during pretraining
  - **Sequence-level tasks**
    - Example: classifying a sequence as positive/negative (detection)

DNABERT: Fine-tuning Applications

- DNABERT is fine-tuned on three tasks (sequence-level):
  1. Prediction of promoters
  2. Prediction of transcription factor binding sites
  3. Prediction of splice sites

DNABERT: Fine-tuning Prediction of Promoters

- **Promoter:** DNA sequence indicating where the **transcription** should start
  - Found before the sequence to transcript (coding sequence)
  - Indicates where the “cell machinery” (polymerase) will initiate the transcription
- **Promoter examples**
  - RNA polymerase binding site, e.g., TATA box, TSS...
  - Length: 100-1000 DNA bases (“characters”)
- **Predicting promoters**
  - Challenging bioinformatics problem!

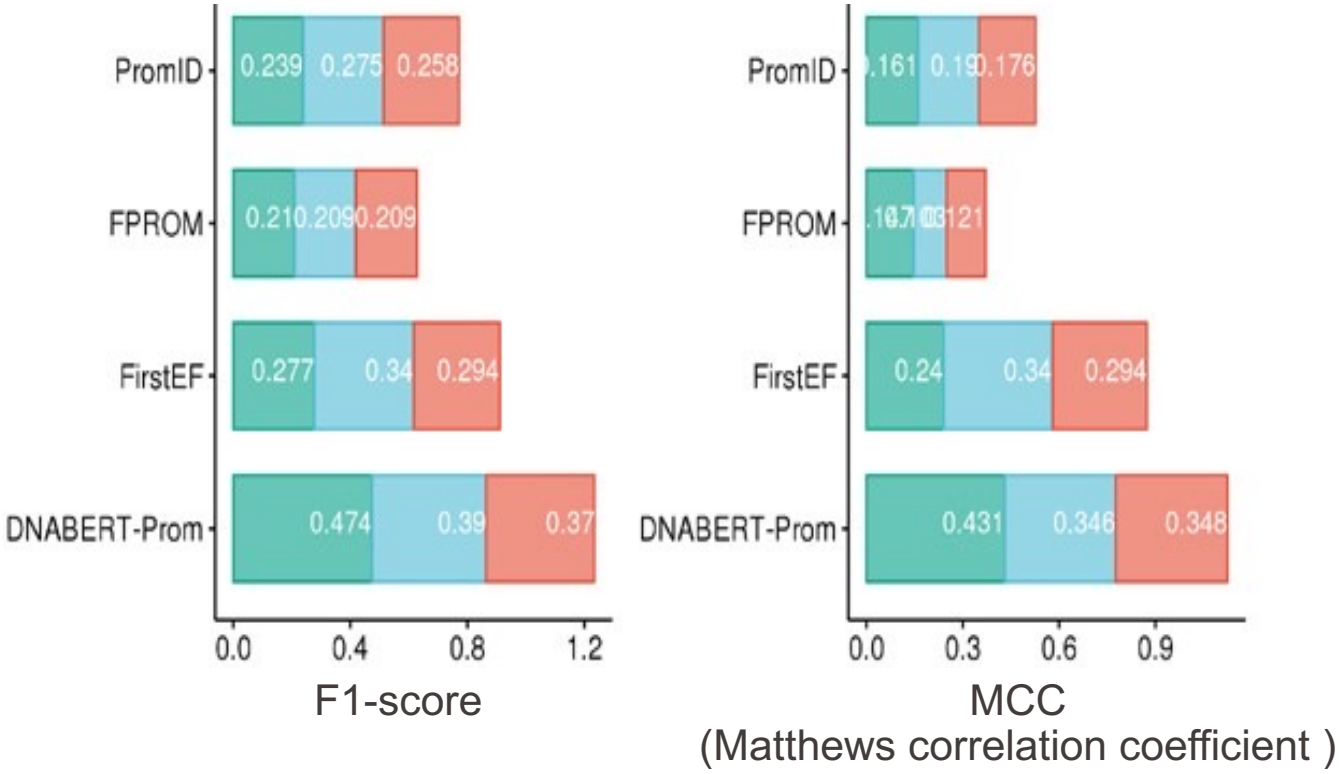


DNABERT: Fine-tuning Prediction of Promoters

- Binary classification of sequences (detection)
- Specific data used to fine-tune DNABERT
  - Positive samples: promoter sequences (from Promoter Database)
  - Negative samples: random sequences outside promoter regions
    - Random sequences are not enough
    - Use of random sequences containing similar motifs (TATA)
  - Selecting “difficult” negative sequences helps DNABERT to learn less obvious features to discriminate positive/negative samples

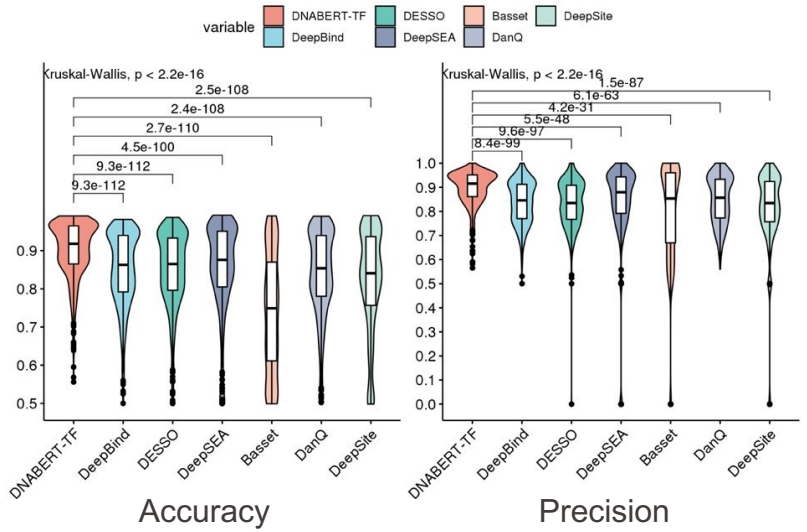
DNABERT: Fine-tuning Prediction of Promoters

- DNABERT significantly outperforms other models in identifying promoter regions

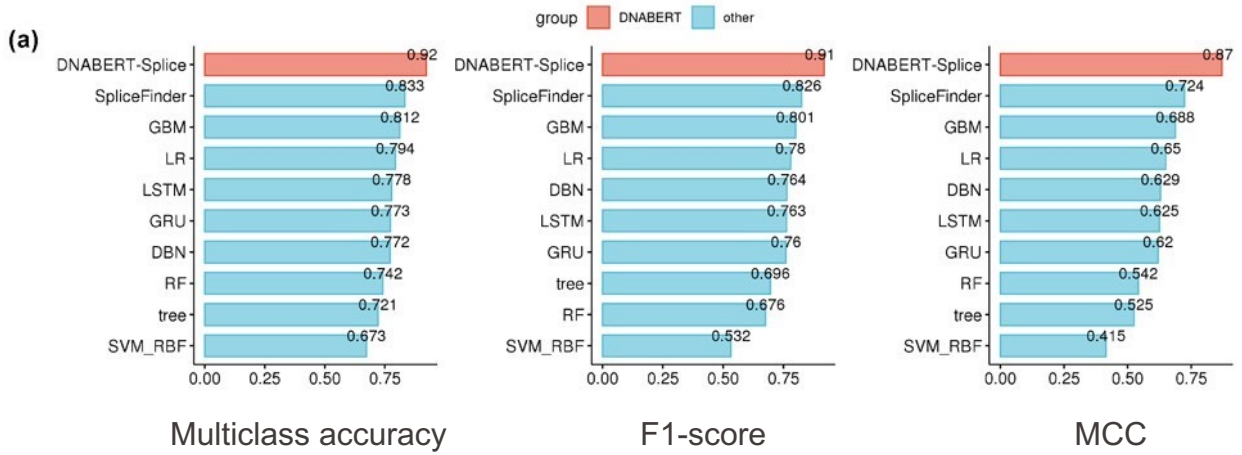


DNABERT: Fine-tuning Other Finetuned Models

- **Same idea as promoter prediction**
  - Sequence-level tasks
  - Binary classification (positive/negative) to detect specific sites
  - Different specific data for transcription factor binding sites & splice sites
  - **DNABERT is also outperforming other existing models !**



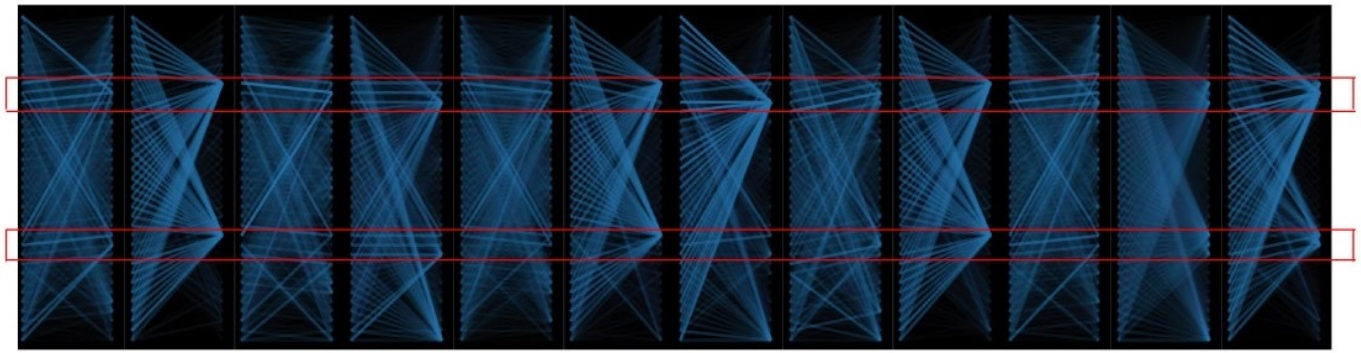
2. Prediction of transcription factor binding sites

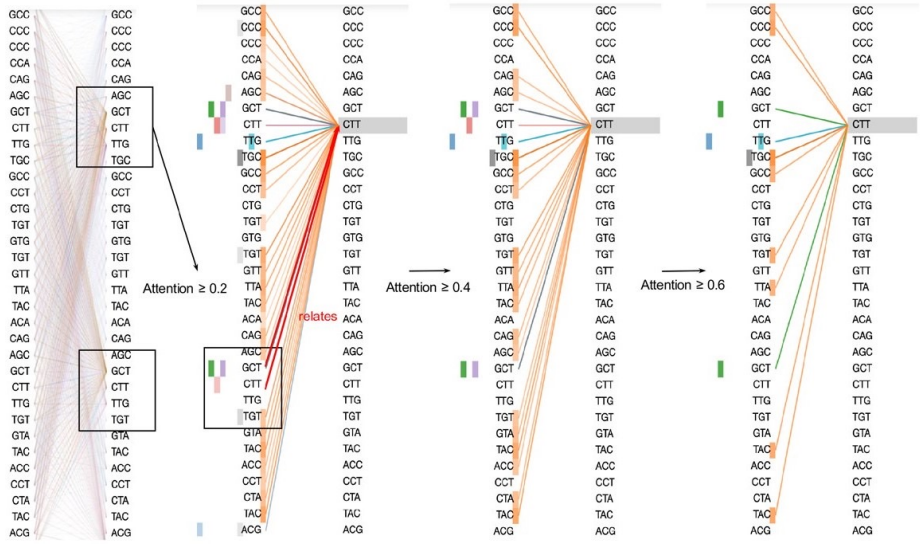


3. Prediction of splice sites

DNABERT

- **Interested in more details ?**
  - Check the publication: https://academic.oup.com/bioinformatics/article/37/15/2112/6128680





ScGPT

Cui etl al. scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI. *bioRxiv* 2023

But Before ...

Why single-cell genomics?

Analogy to Bee Colony

All bees work together for a colony to function

Every bee has a role

Gathering nectar

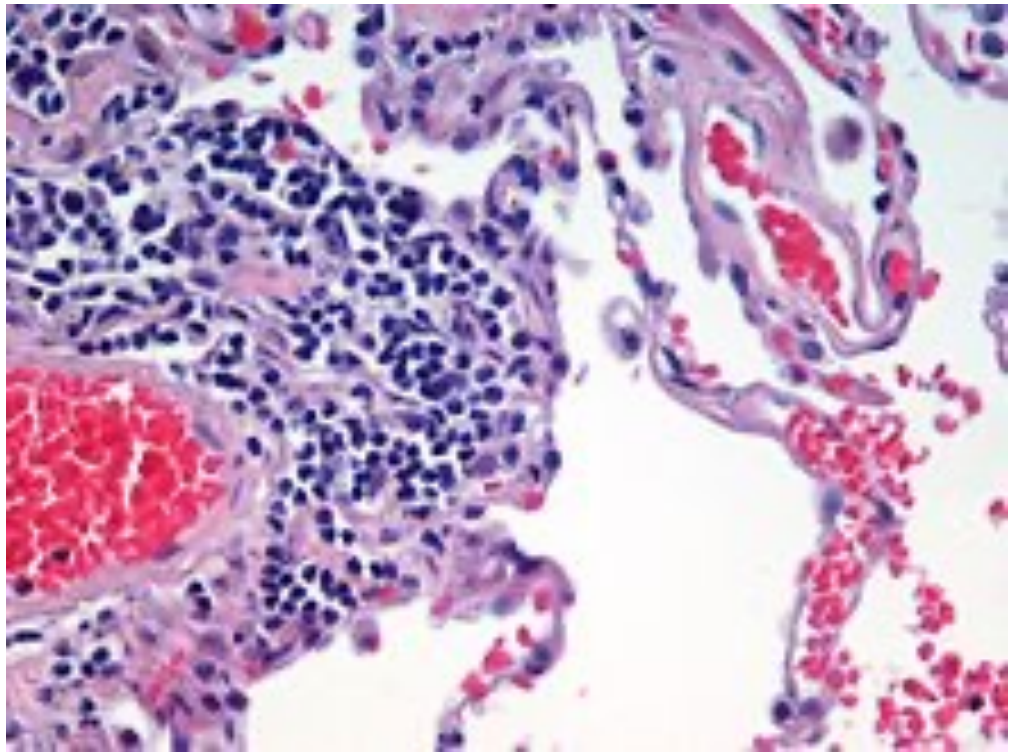
Cleaning the hive

Guarding the hive

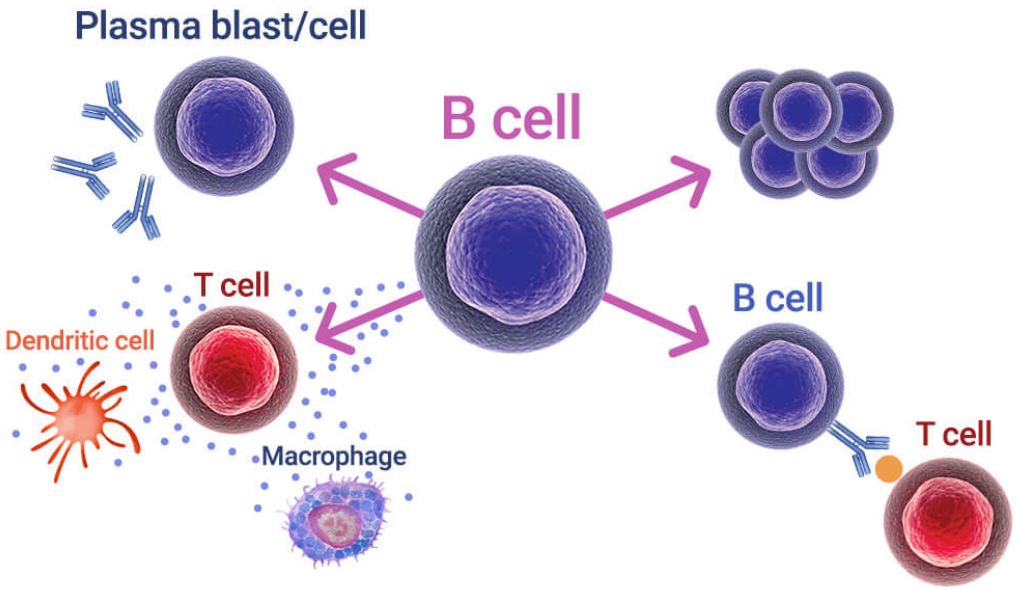
Protecting the queen

Our Cells Are Very Heterogenous!

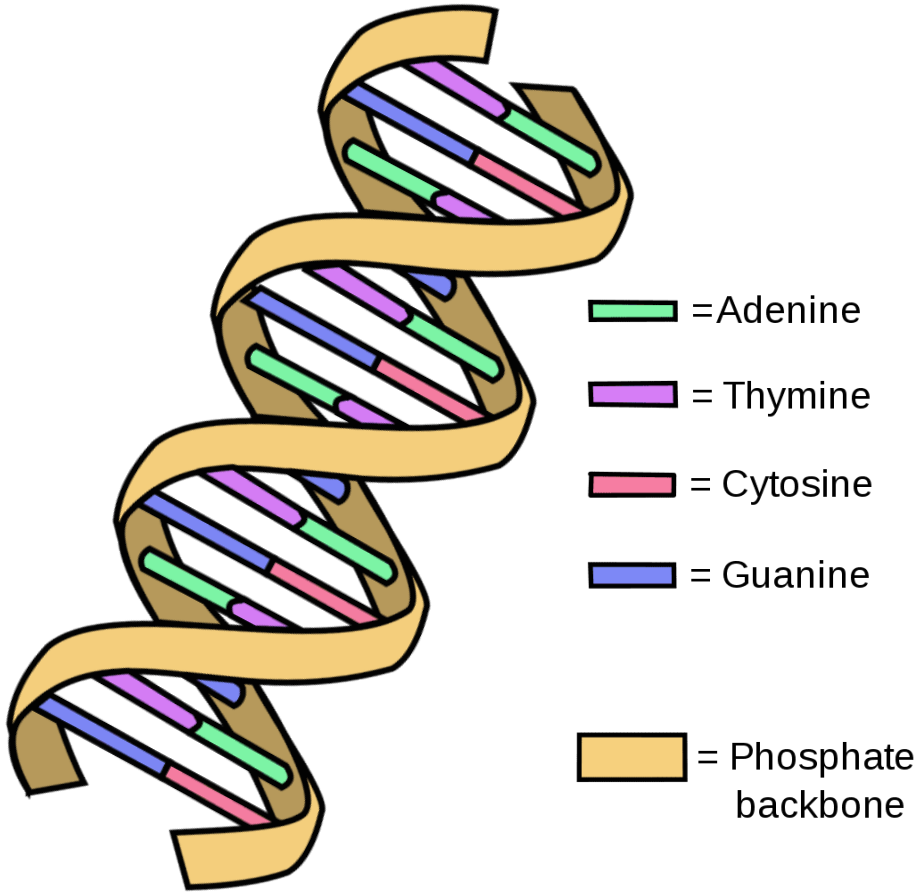
All ~~bees~~ cells work together for a ~~colony~~ tissue to function



Every ~~bee~~ cell has a role



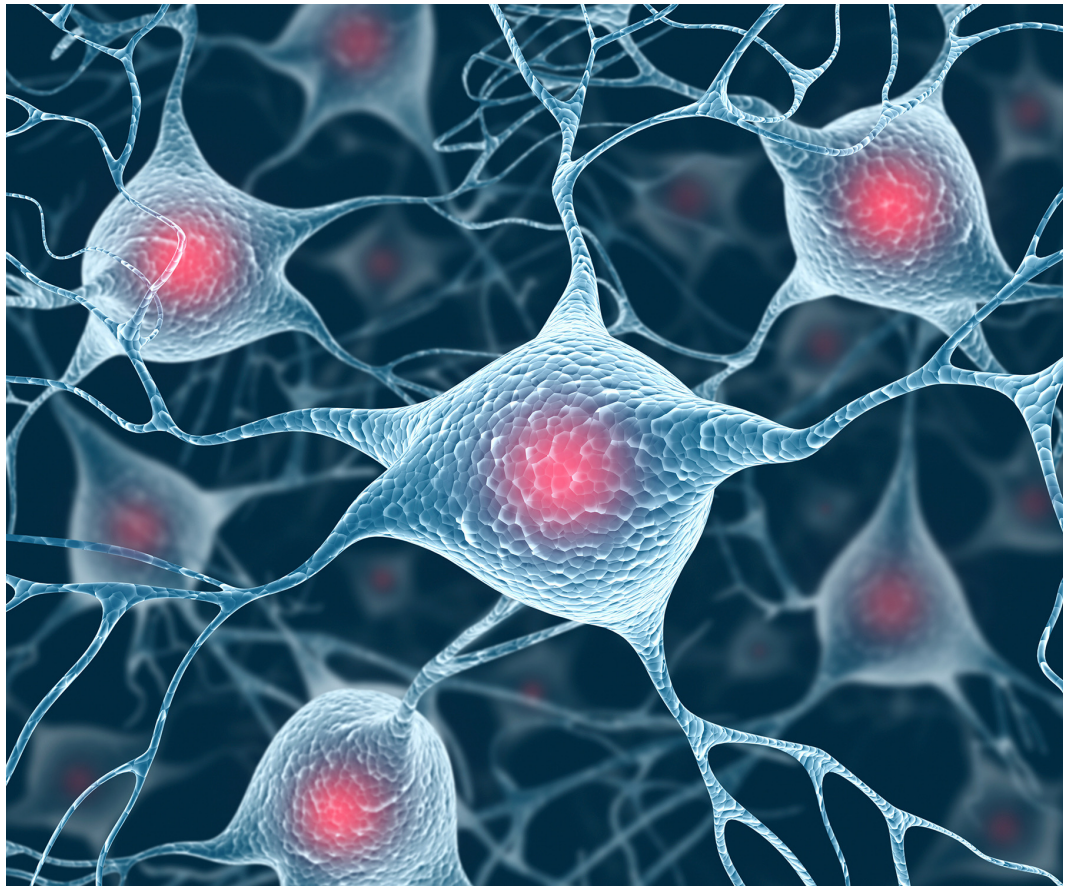
DNA Is Shared Among Cells



DNA is the same in all our cells!

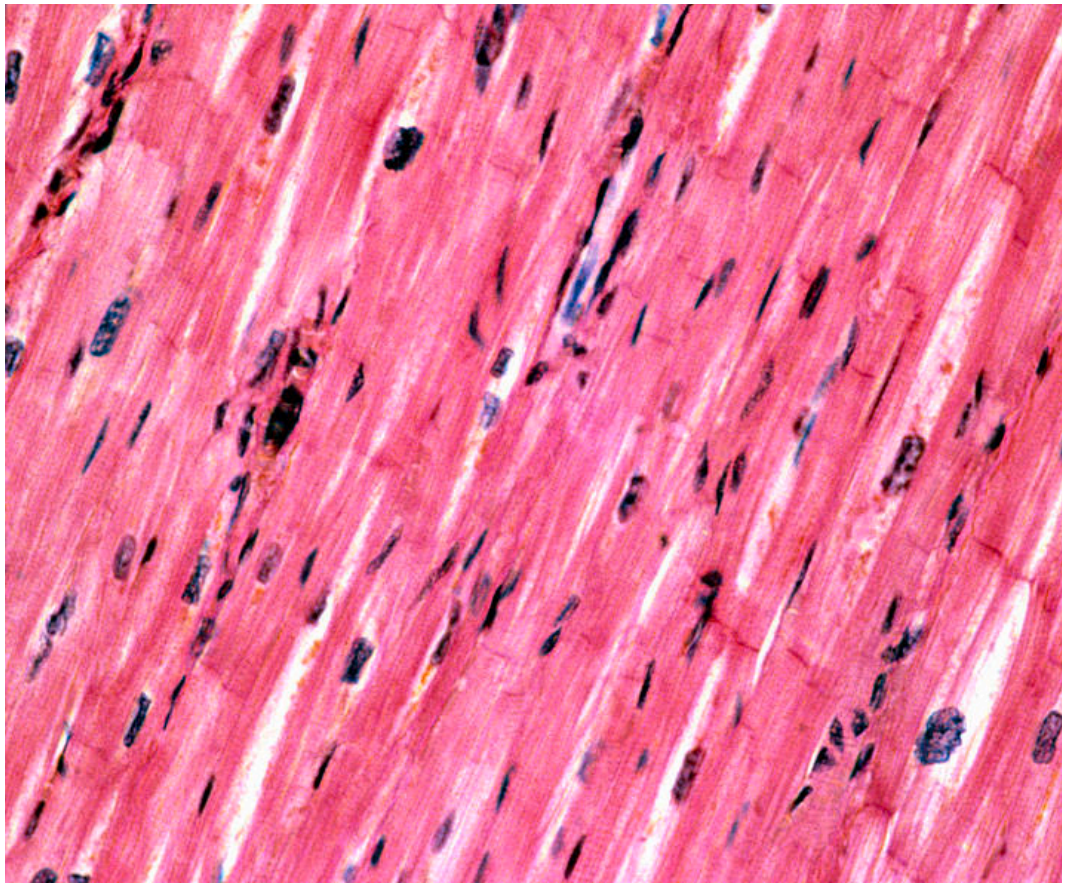
But, what then makes our cells so different?

Brain cells



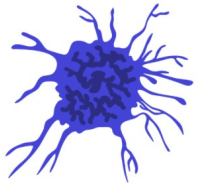
vs

Muscle cells

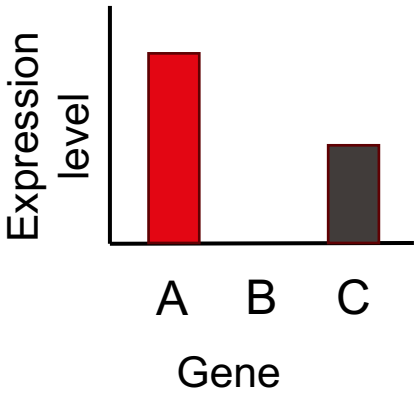


Different Gene Expressions Define Cell Function and Morphology

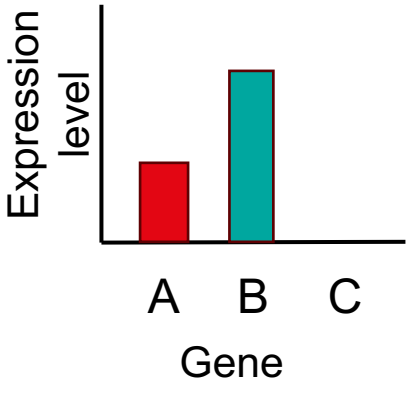
Different genes are differently expressed across cells!



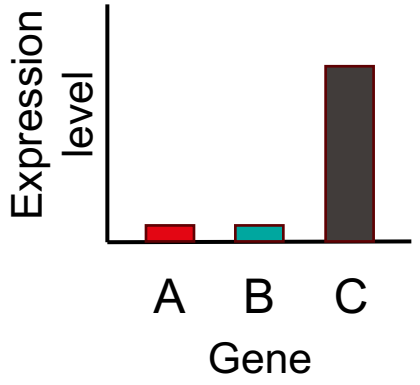
Dendritic cell



Macrophage



B-cell



Can We Measure This?

Yes! 

How?

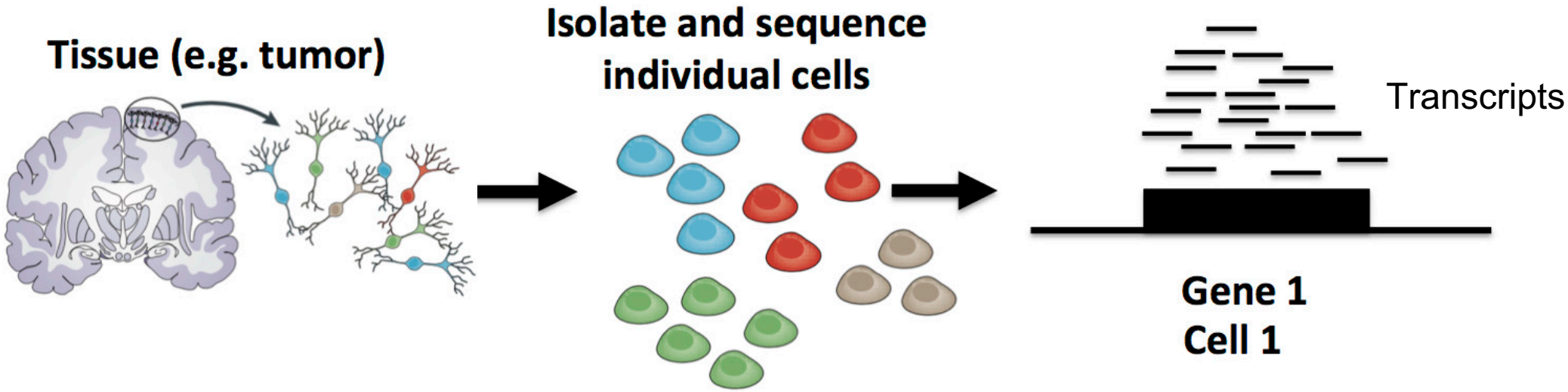
Using single-cell RNA sequencing!

Method of the Year 2013

Methods to sequence the DNA and RNA of single cells are poised to transform many areas of biology and medicine.

Single-cell sequencing provides resolution on a level of individual cells.

Single-Cell RNA Sequencing (scRNA-Seq)



Counts represent number of transcripts of a gene in a cell.

	Cell 1	Cell 2	...
Gene 1	18	0	
Gene 2	1010	506	
Gene 3	0	49	
Gene 4	22	0	
...			

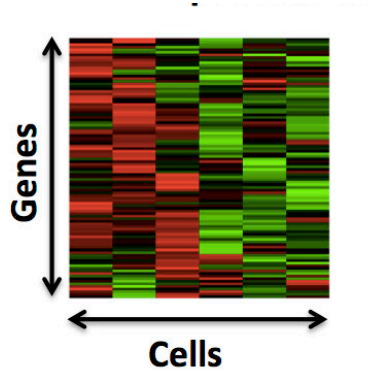
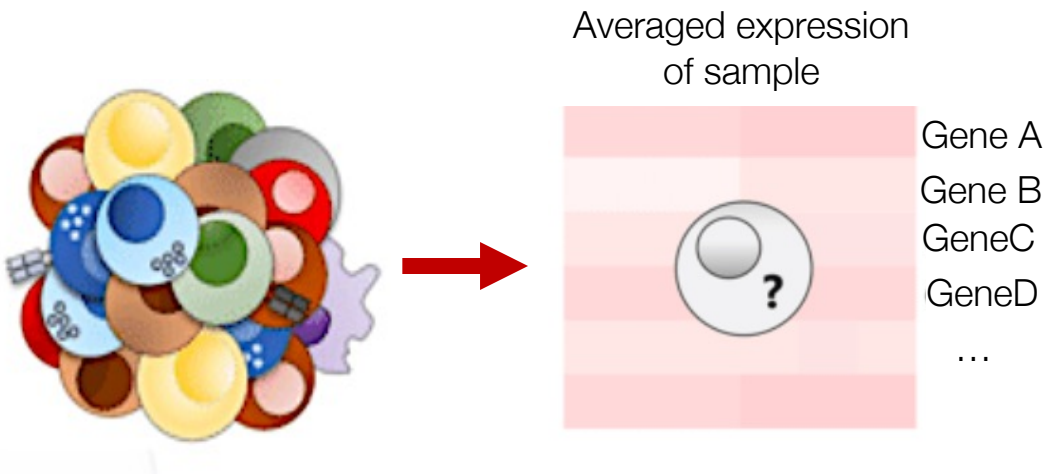


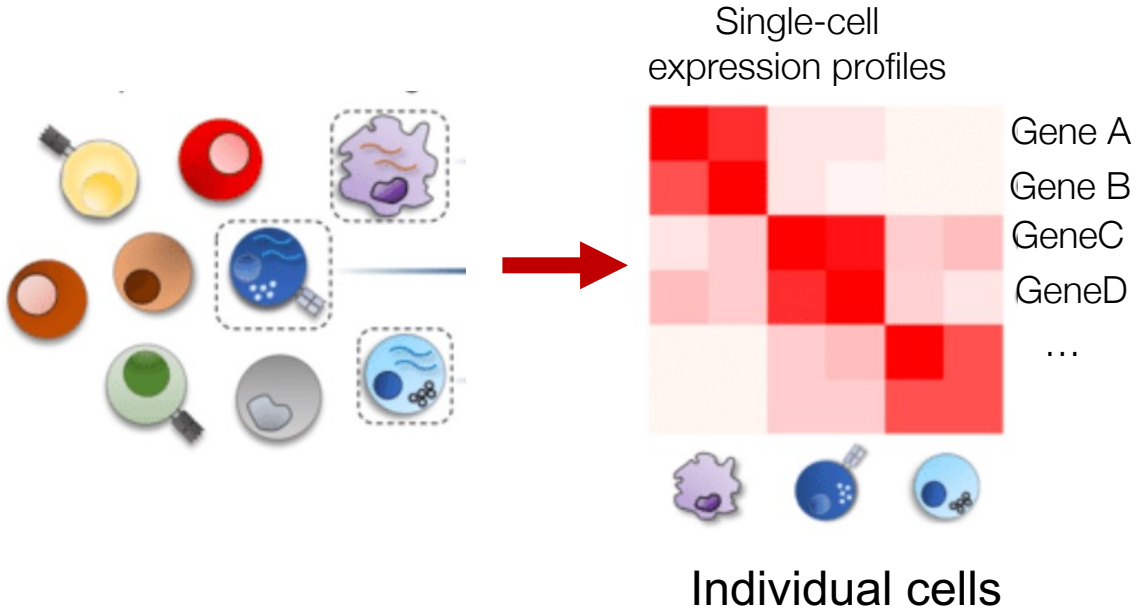
Figure from rna-seqblog

Bulk vs Single-Cell Sequencing

Bulk sequencing



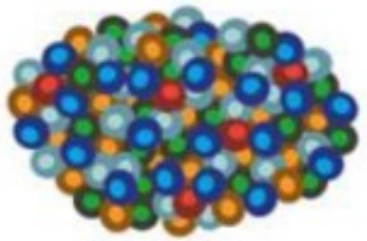
Single-cell sequencing



Bulk vs Single-Cell Sequencing

Bulk sequencing

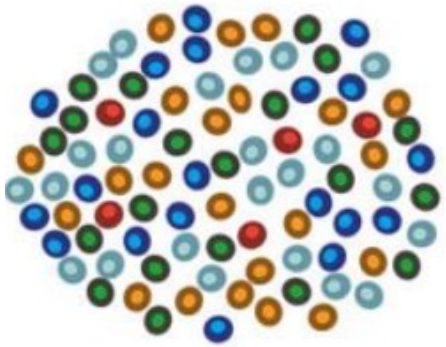
Bulk tumor



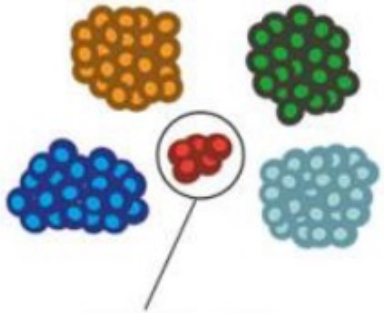
Inability to resolve cell populations

Single-cell sequencing

Dissociated tumor



Identification of different cell populations



Population of interest

Bulk vs Single-Cell Sequencing

Bulk sequencing

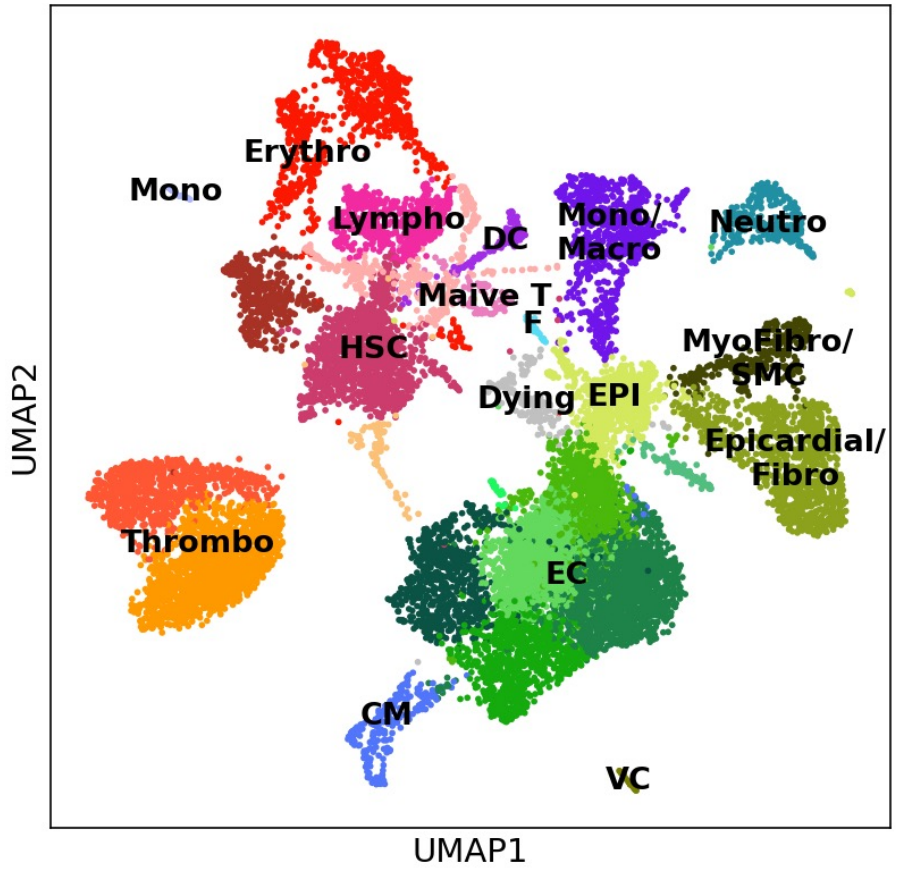
Single-cell sequencing

What Can Be Measured

- Single-Cell RNA Sequencing (scRNA-seq):
  - Measurement of gene expression in individual cells.
- We can also measure other aspects of cells!
  - Proteome, epigenome, spatial organization, metabolome…

Single-cell RNA Data Visualization

- Raw single-cell RNA data can be visualized as a UMAP



Key ideas:

- Each point represents a cell
- Each cluster represents a cell type
- UMAP projects scRNA data from $\mathbb{R}^{N_{gene}}$ to $\mathbb{R}^2$
- Think PCA but non-linear!

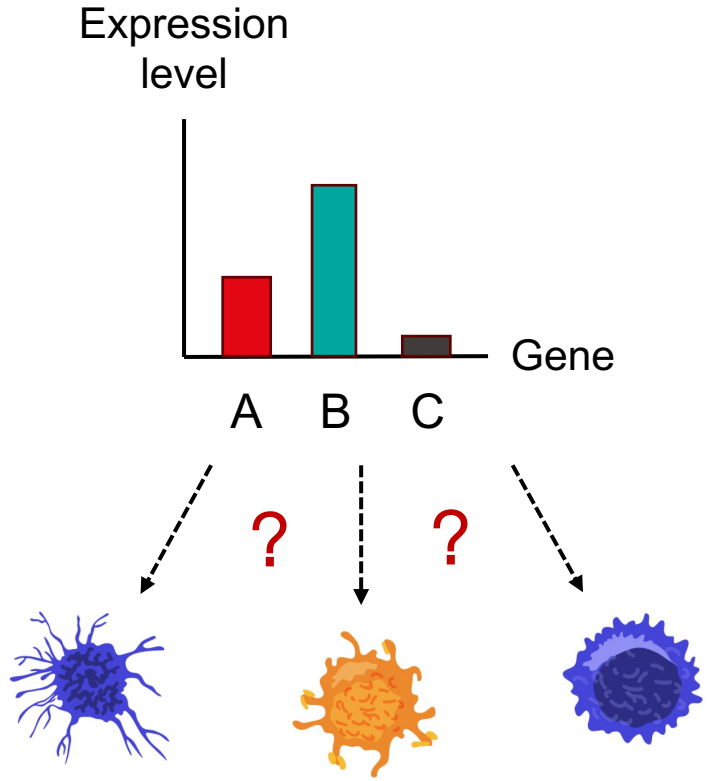
McInnes, Leland; Healy, John; Melville, James. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* 2018.

Challenges

Noise

- Technical noise, e.g. variations in amplification and sequencing errors.
- Inherent biological variability between cells of the same type.

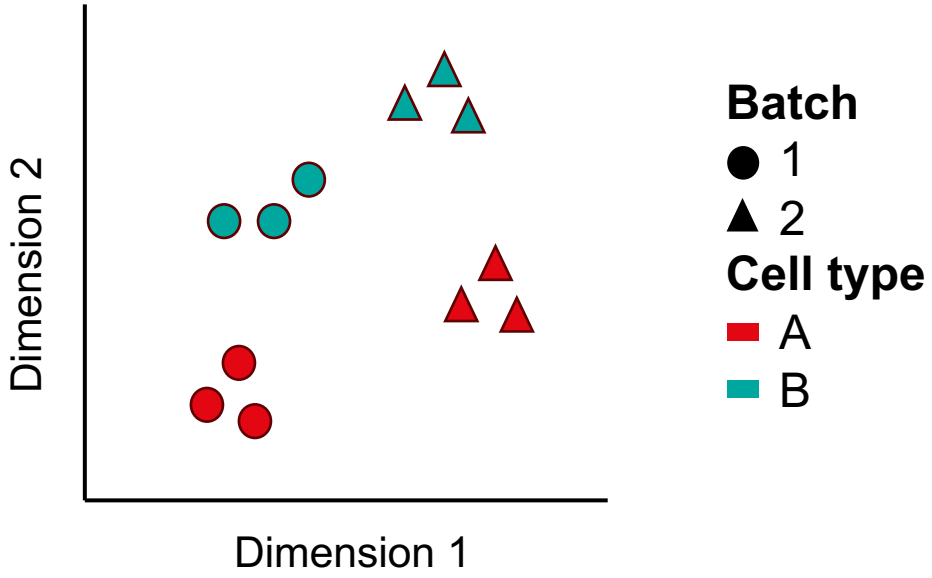
Unannotated data



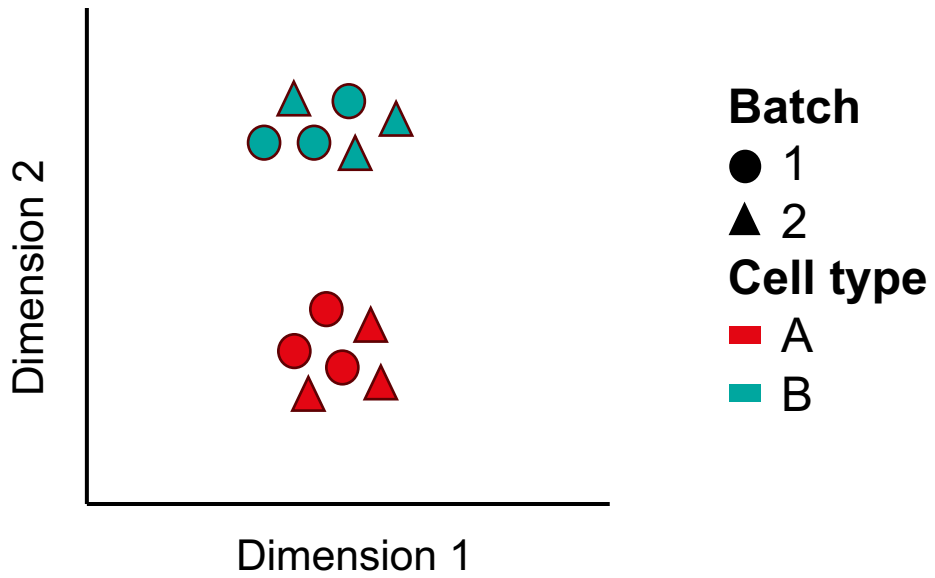
Challenges: Batch Effects

Batch effects: systematic and non-biological variations in data that arise from the technical or experimental processes

Observed data



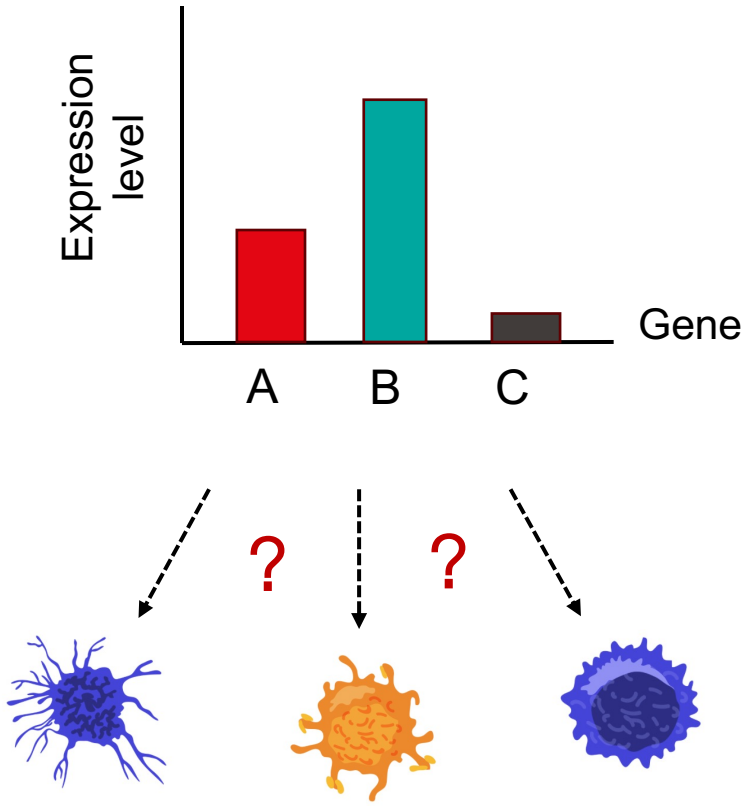
"True" underlying process



What Computational Problems and Tasks Arise When Dealing with Single-cell Data?

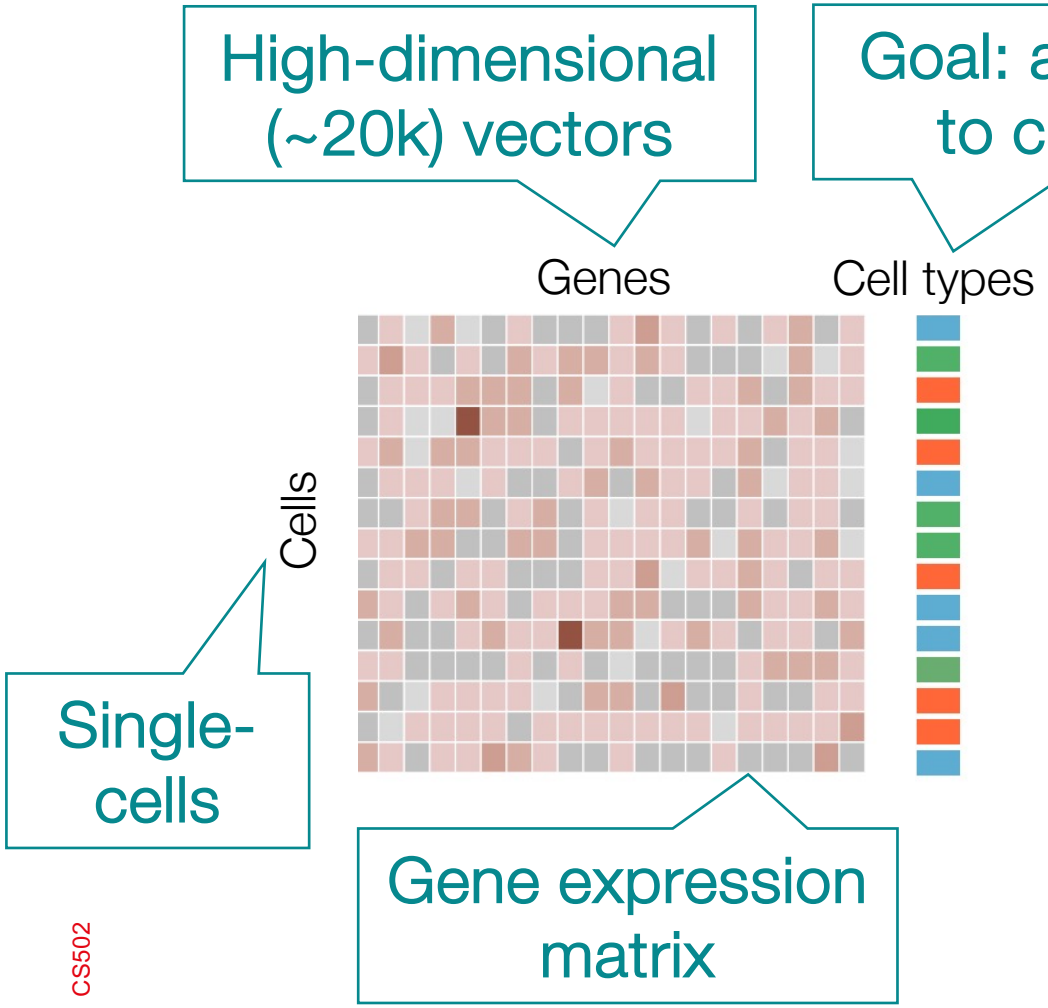
Task: Cell Type Annotation

Challenge: Unannotated datasets



Goal: Given gene expression profiles of cells, assign cells to different cell types

Task: Cell Type Annotation



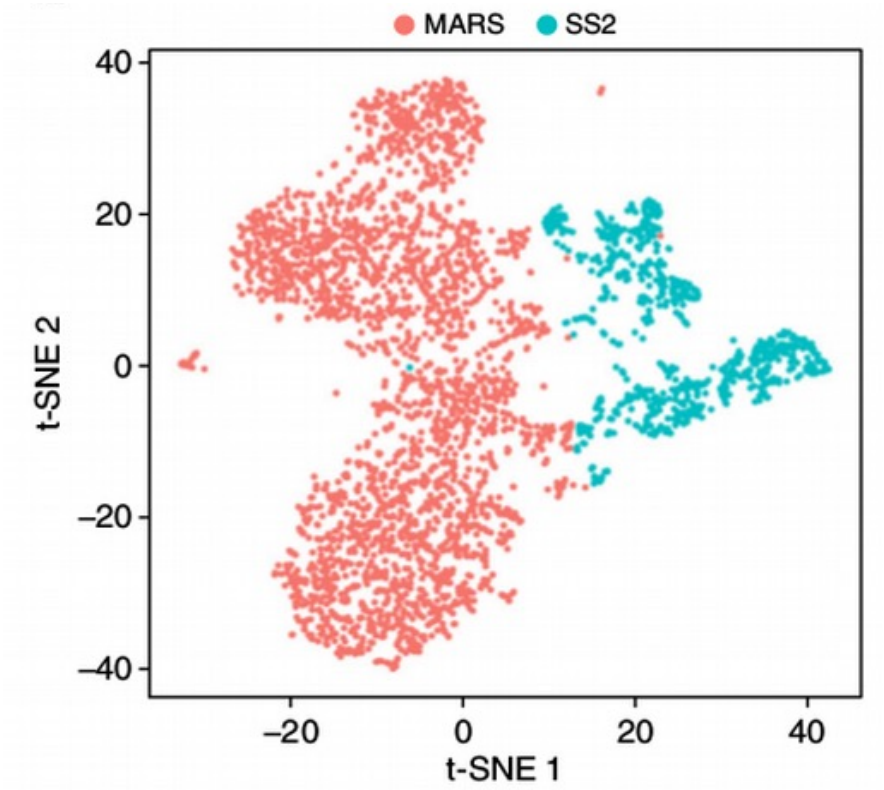
Goal: Given gene expression profiles of cells, assign cells to different cell types

- Can be unsupervised, supervised, semi-supervised

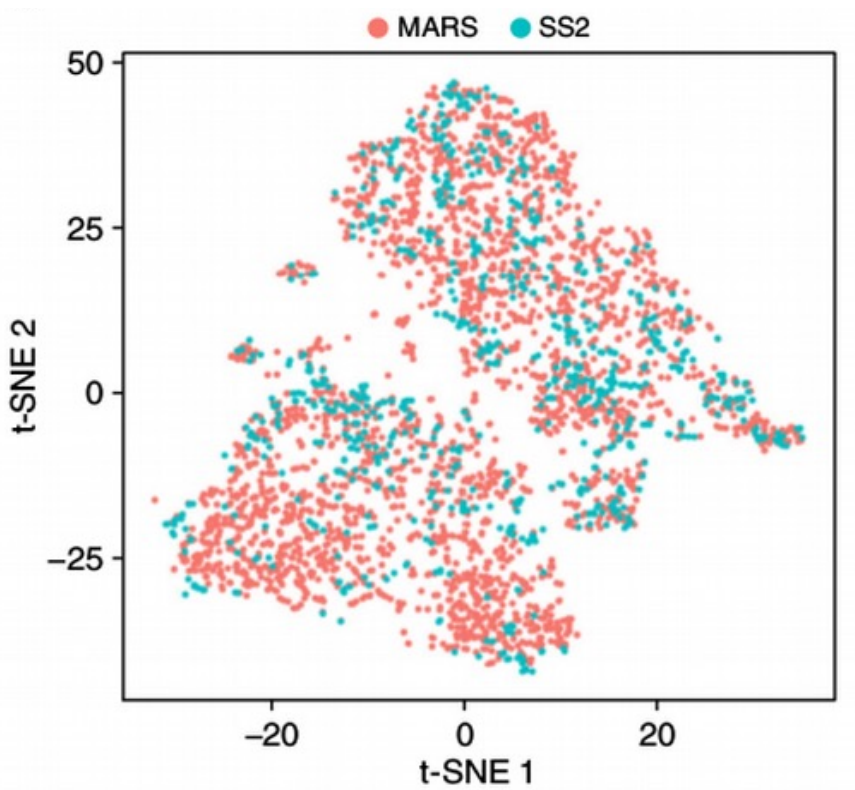
Task: Batch Correction

Challenge: Batch effects

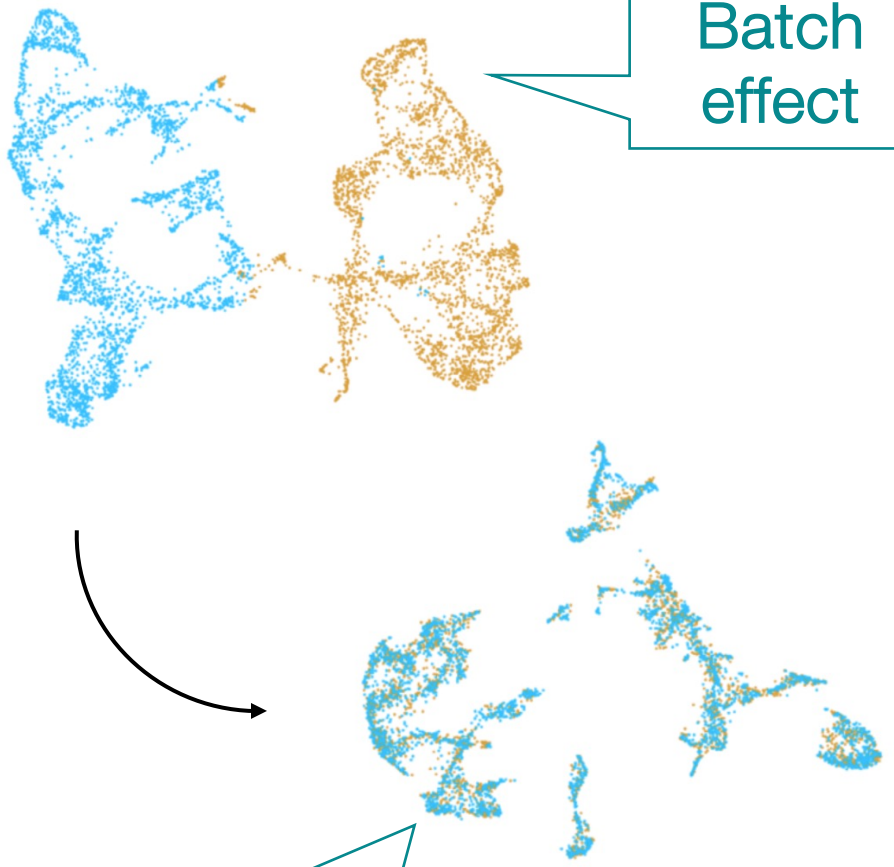
Input datasets



Batch-corrected datasets



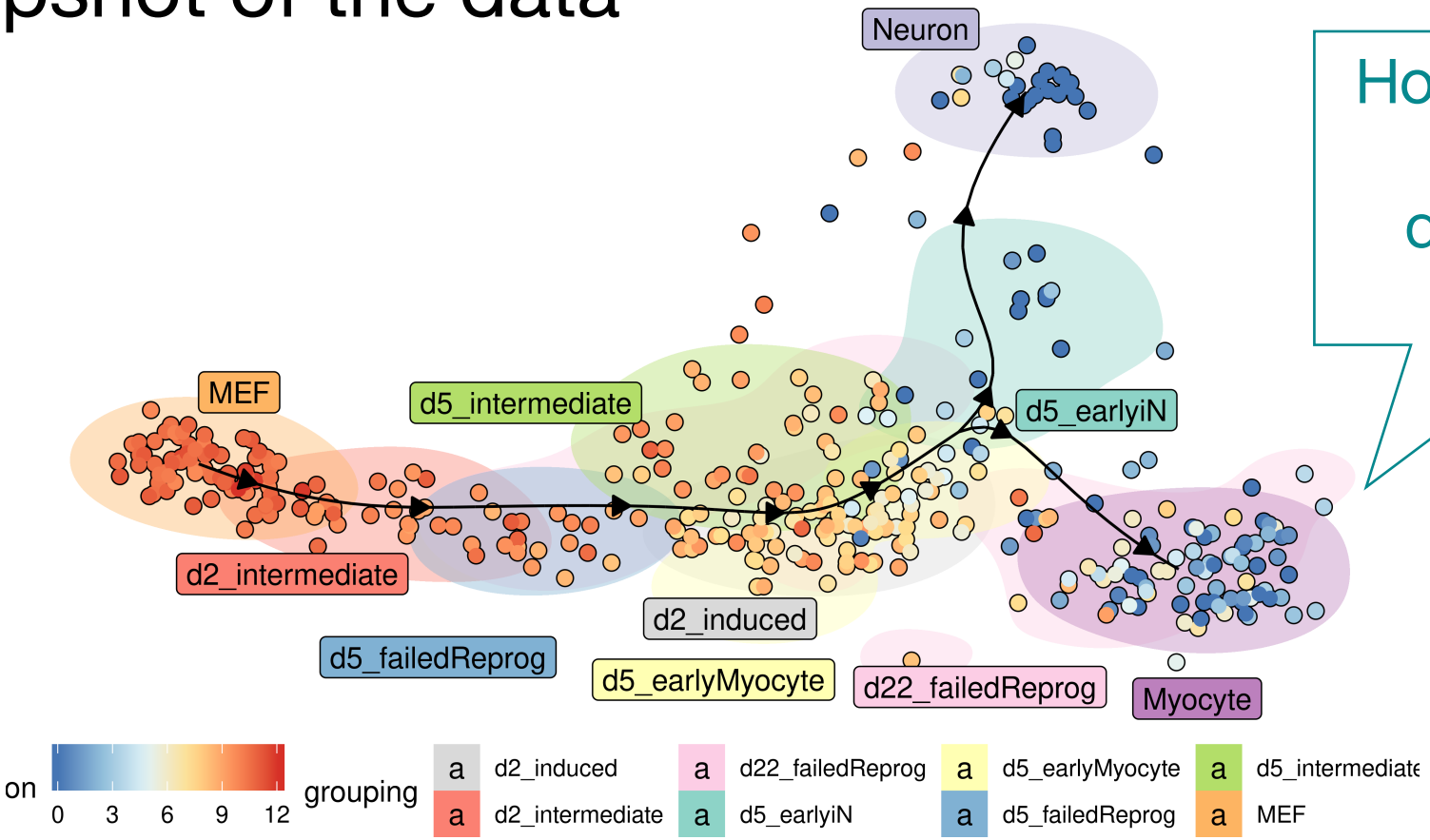
Batch Correction



Goal: Given two or more datasets from different experiments, remove technical variation and retain only biological variation

Trajectory Inference

Challenge: Single-cell data provides only single snapshot of the data



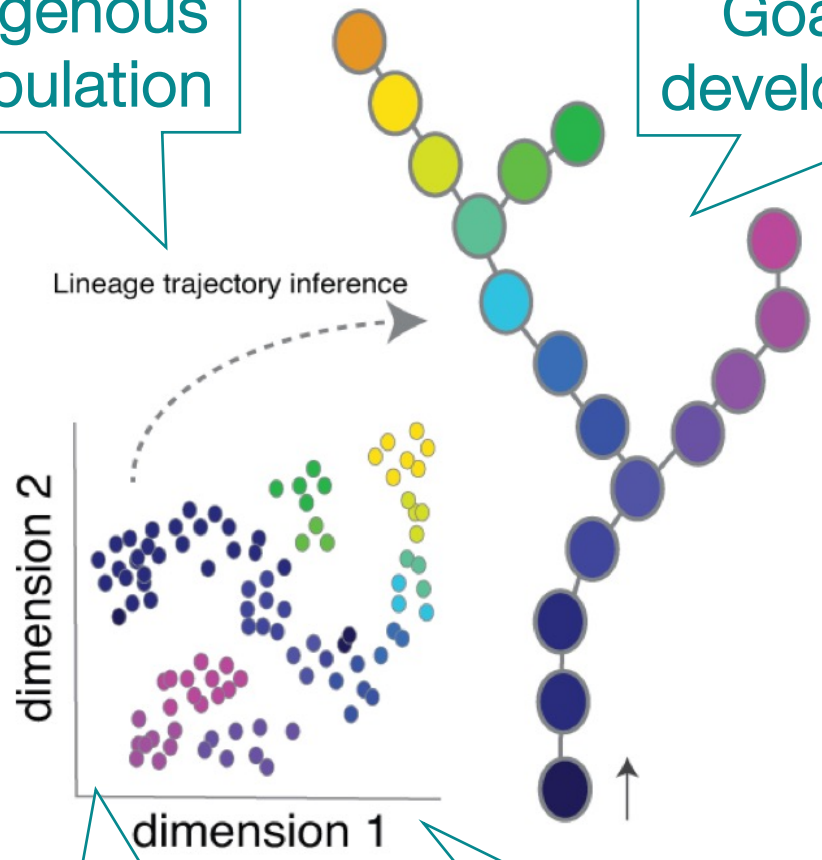
How to order cells along developmental trajectories?

Figure from https://dynverse.org/dyno/

Trajectory Inference

Heterogenous cell population

Goal: construct developmental path



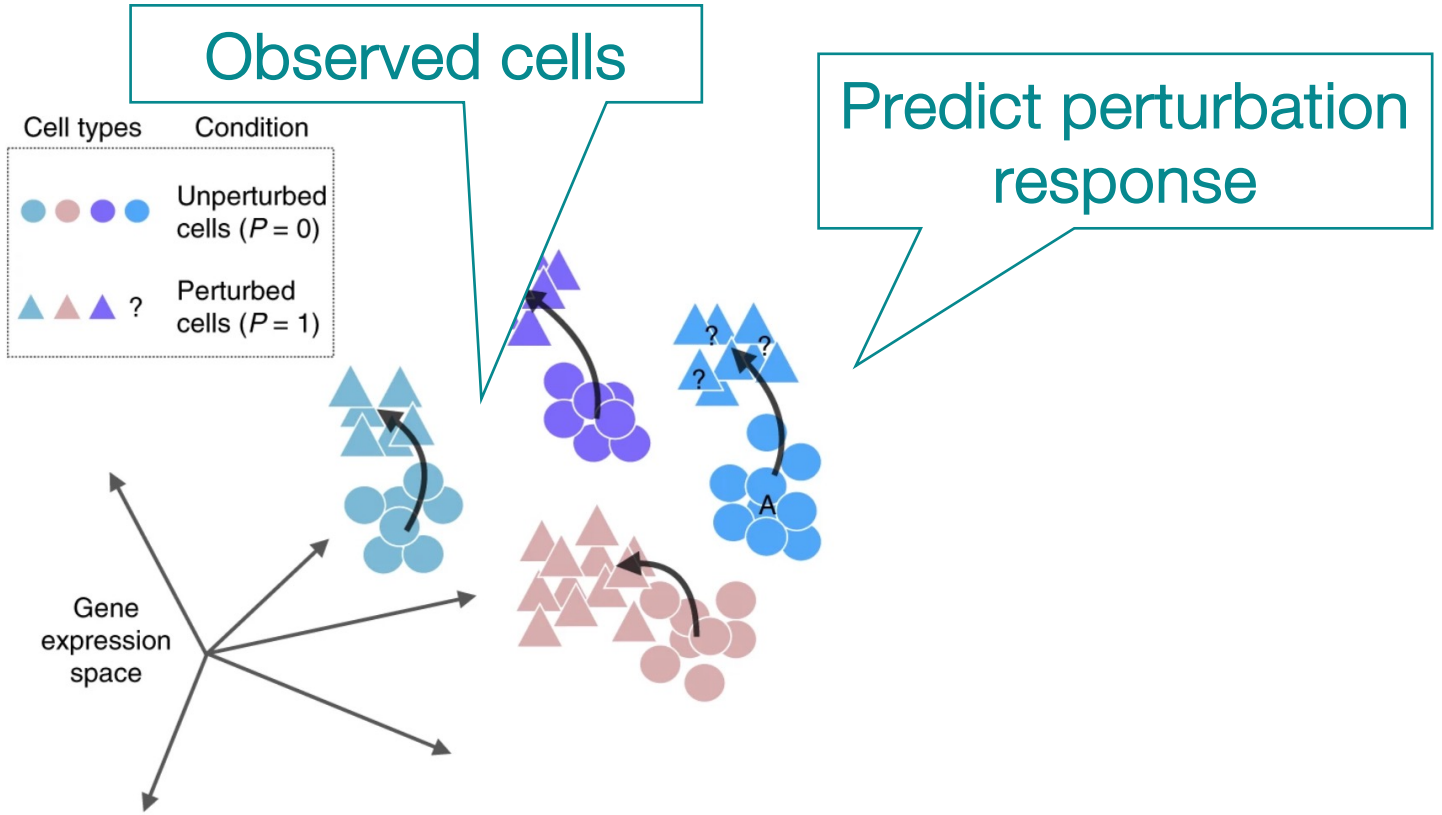
Single-cells

Reduced dimension

Goal: Given gene expression profiles of cells, the goal is to infer their order along the developmental process

Perturbation Response Prediction

Challenge: Experimentally unfeasible to perturb all combinations of genes and measure effects



Perturbation Response Prediction

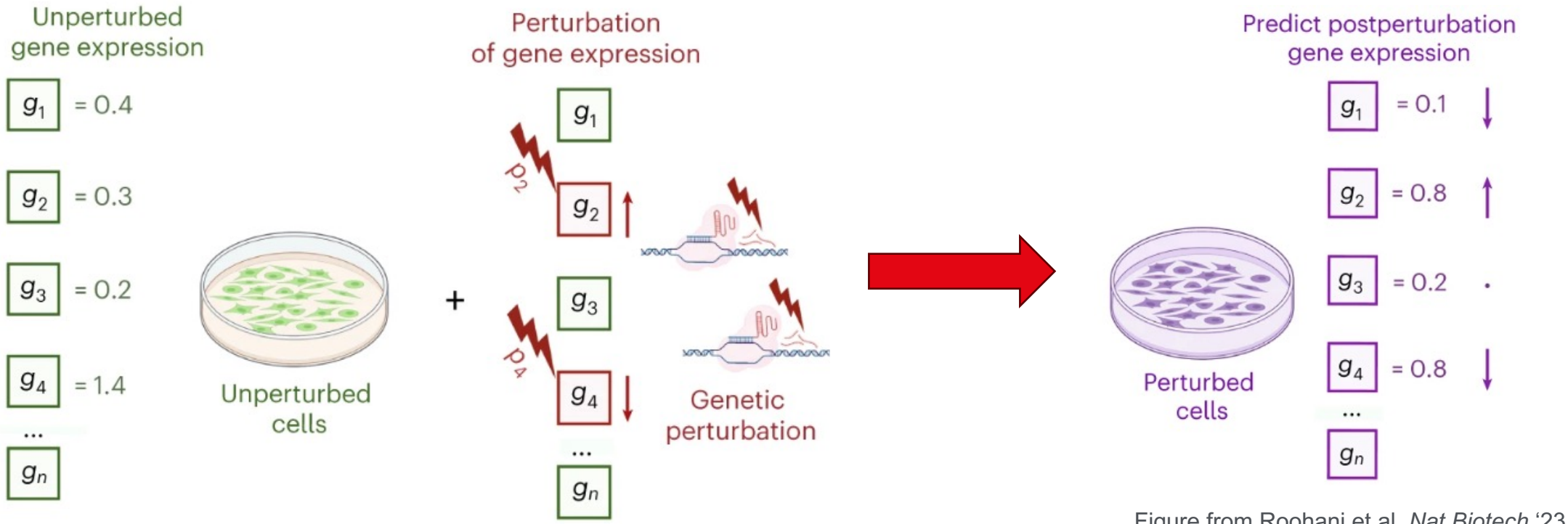


Figure from Roohani et al. *Nat Biotech* '23

Goal: Given a set of observed perturbations, predict the perturbation response of new gene combinations

Ready to discuss scGPT!

Cui etl al. scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI. *bioRxiv* 2023

scGPT: Single-cell RNA-seq Data

- Single-cell RNA sequencing measures the quantity of messenger RNA (mRNA) produced by each gene

	gene 1	gene 2	gene 3	gene 4	…
cell a					
cell b					
cell c					
cell d					
⋮					

Cell-by-gene matrix X

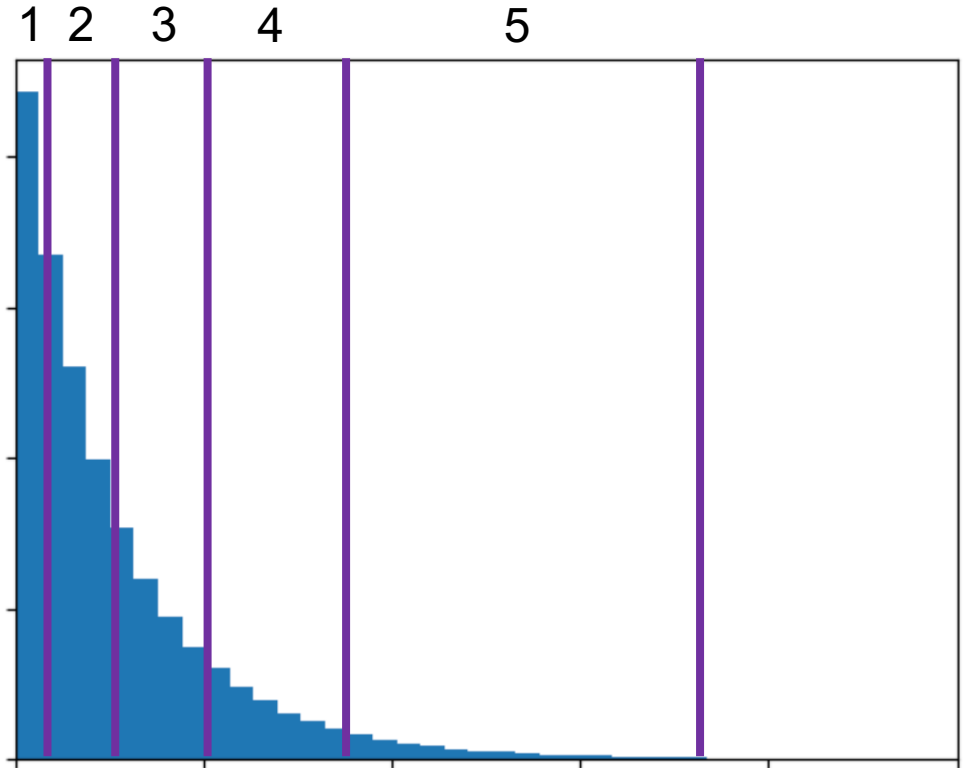
- Dimensions: N_{cell} by N_{gene}
- N_{cell} depends on the experiment
- N_{gene} usually around 10,000 – 25,000
- X_{ij} is the transcript count of gene j in cell i

scGPT: Tokenizing Expression Values

	gene 1	gene 2	gene 3	gene 4	…
cell a					
cell b	2398	75	5401	122	…
cell c					
cell d					
⋮					

Histogram of gene expression levels

Bin the histogram into equal portions
Each bin has the same number of genes
Only non-zero genes are included



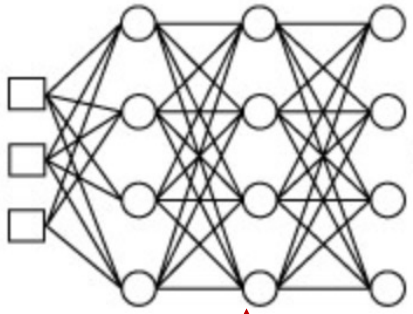
Use bin number to encode expression level

2	5	1	5	…

scGPT: Pretrain & Fine-tune

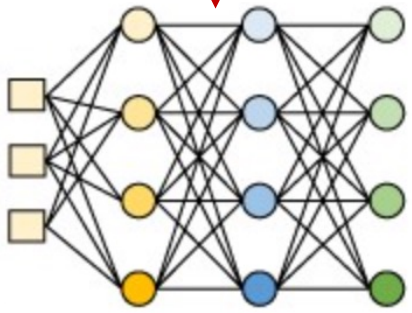
Unlabeled data:
scRNA of 33M cells

CZ CELL×GENE

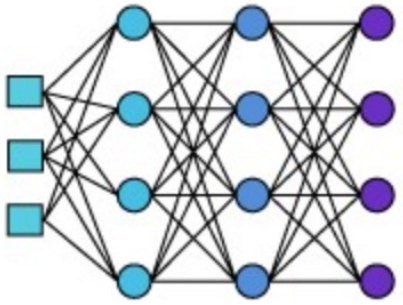


Auxiliary tasks
- **Masked gene prediction**
- **Gene generation by cell prompt**

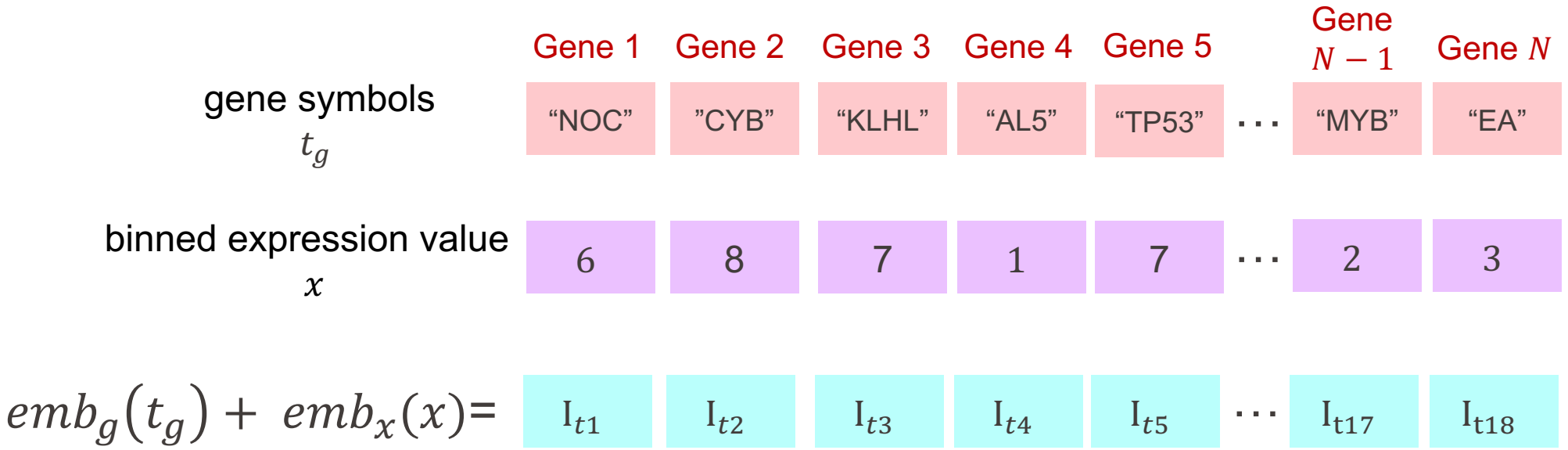
Pretrained model



Fine-tuning using task specific data

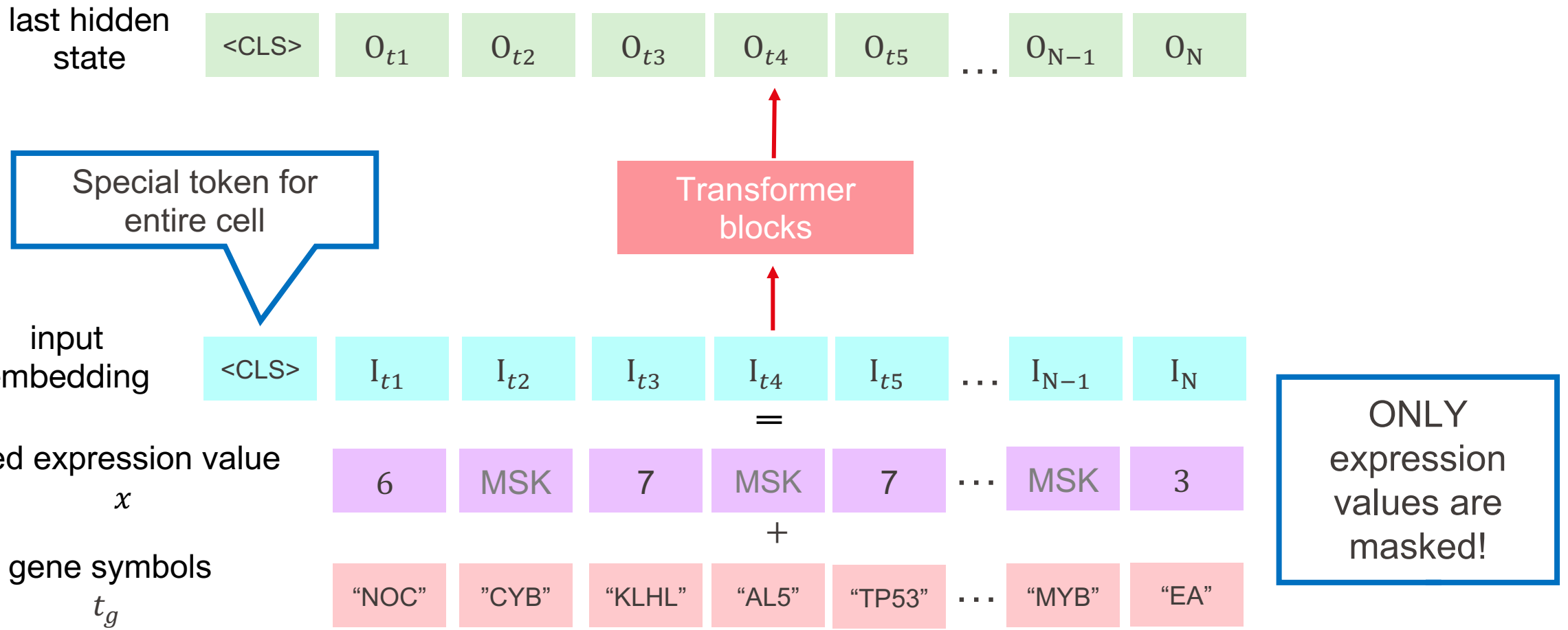


scGPT: Input Embedding



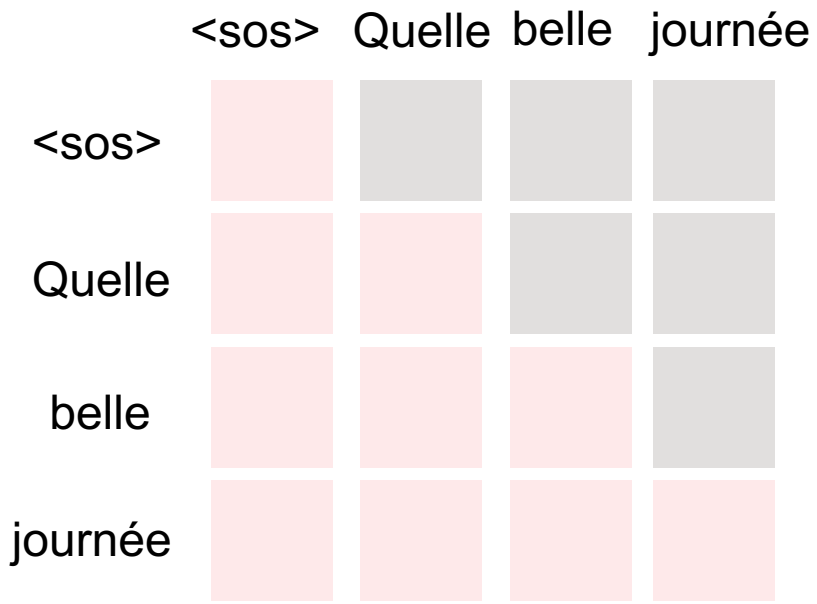
- Separate embedding, $emb()$ for each layer of information
- The final input embedding I is the sum of the embeddings of all inputs

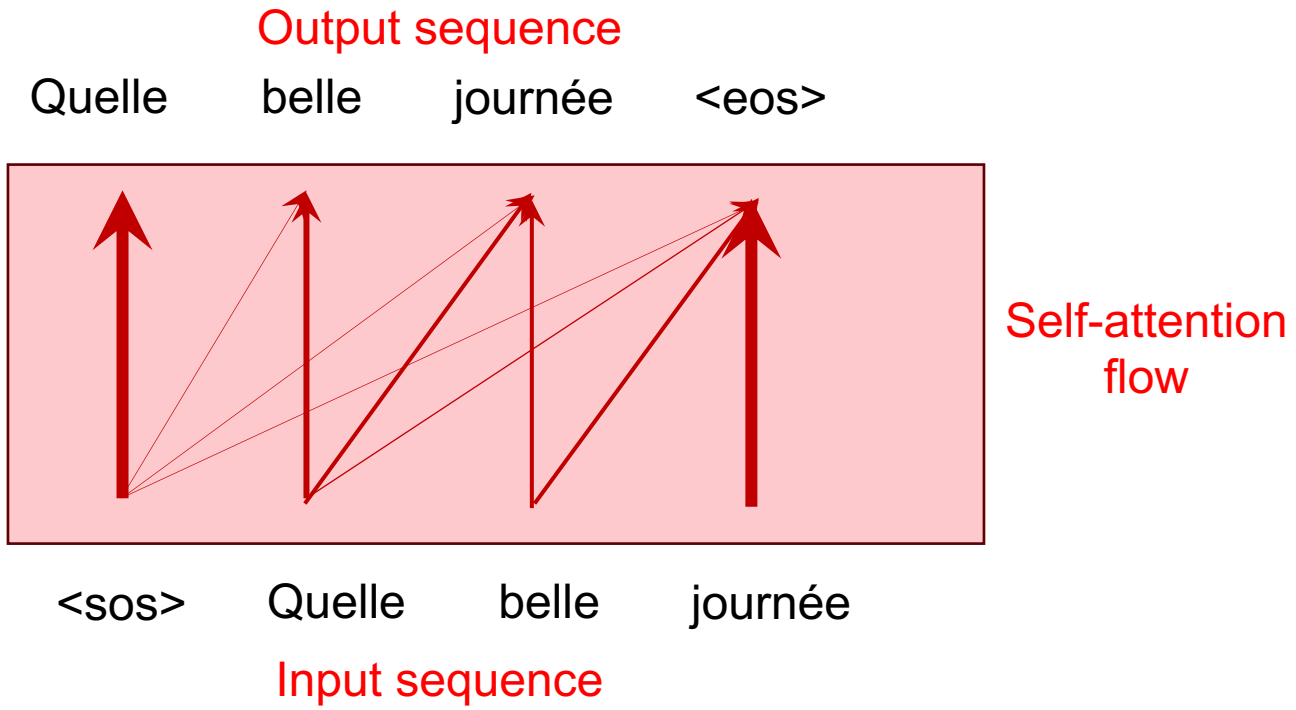
scGPT: Masked Token Prediction Again



scGPT: Attention Masking

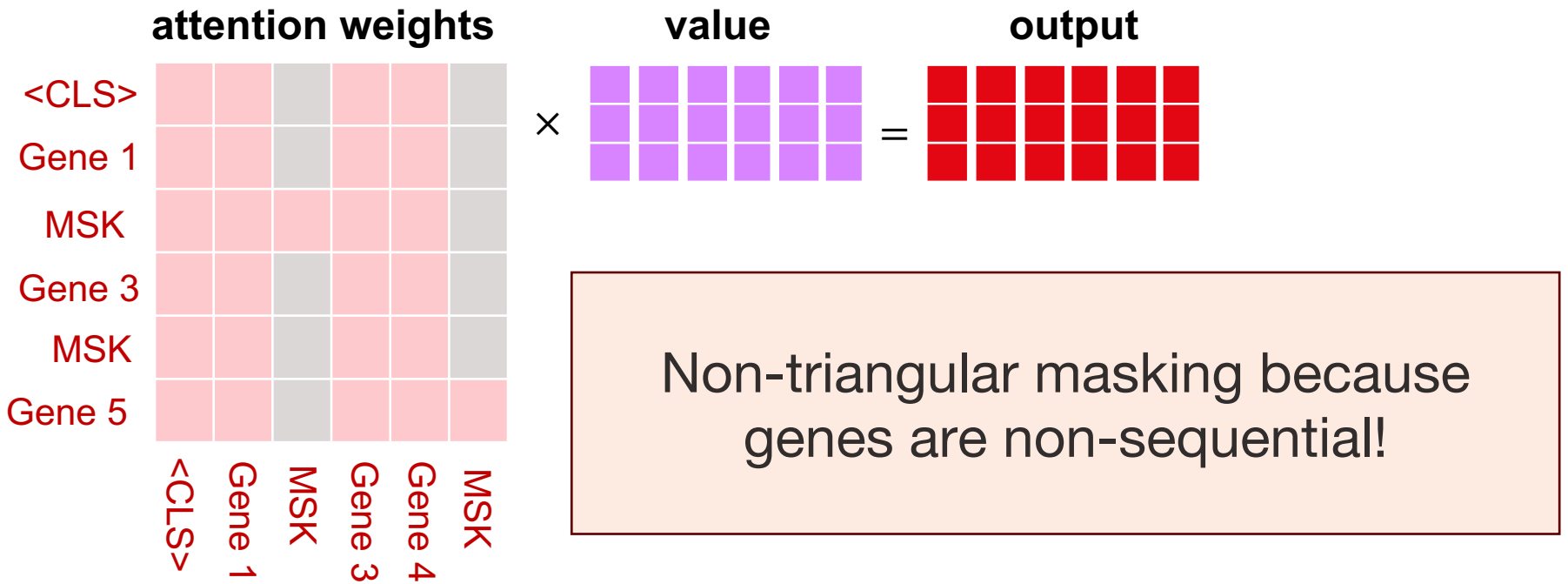
- Recall that in the transformer decoder (e.g. English-to-French translation), upper triangular masking only allows attention to known token when making next-token predictions





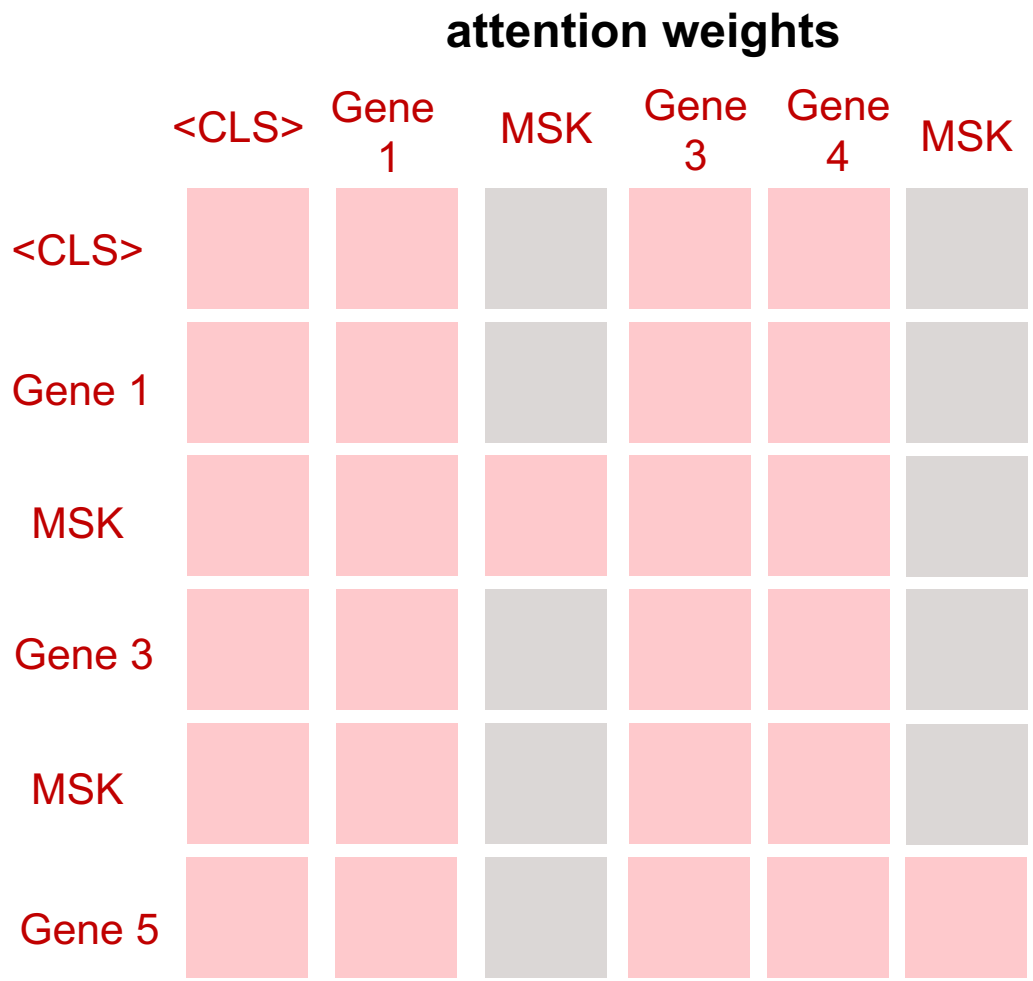
scGPT: Attention Masking

- Similarly, scGPT only allows attention to known genes when generating masked expression levels



Non-triangular masking because genes are non-sequential!

scGPT: Attention Masking



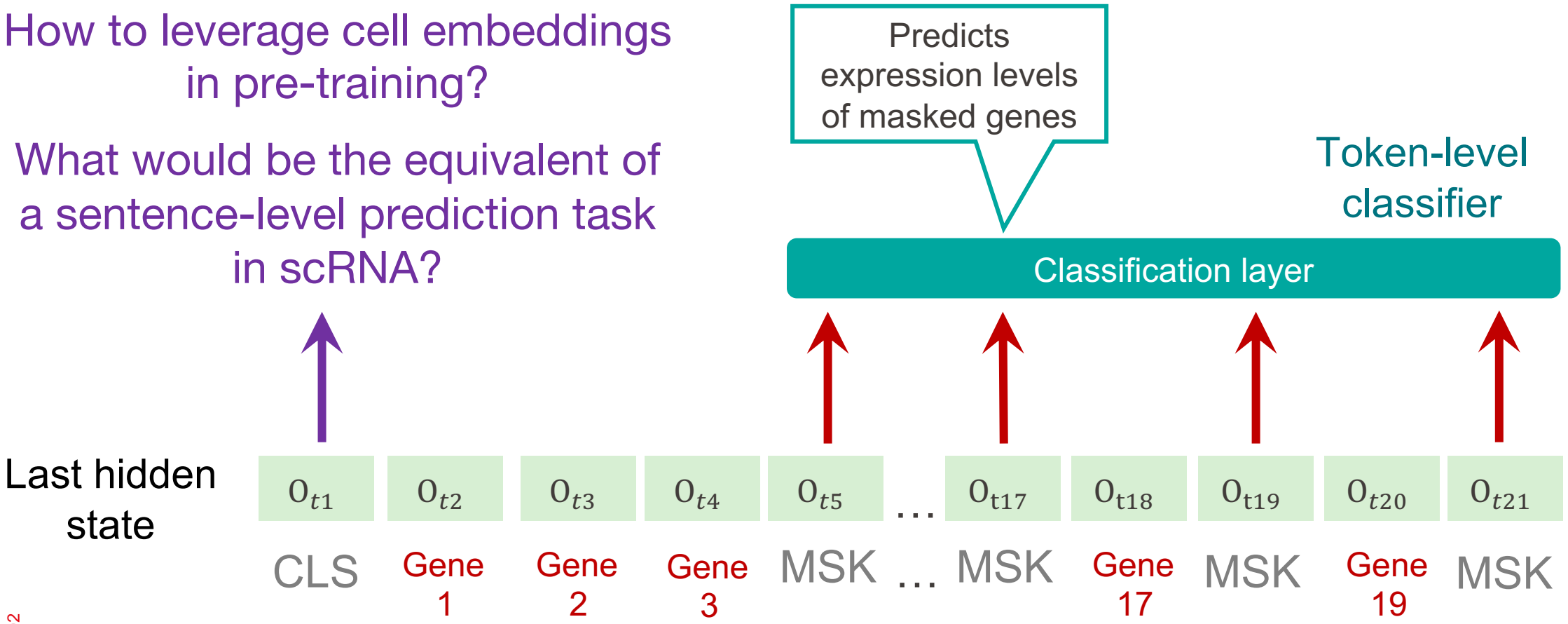
When predicting masked expression of “Gene 2”, attention to the other unknown gene, “Gene 5” is blocked

- When generating <CLS> embedding, attention to both unknown genes is blocked
- Autoregressive model: utilize previous predictions for generating new ones
  - ➢ Genes are gradually unblocked based on confidence scores

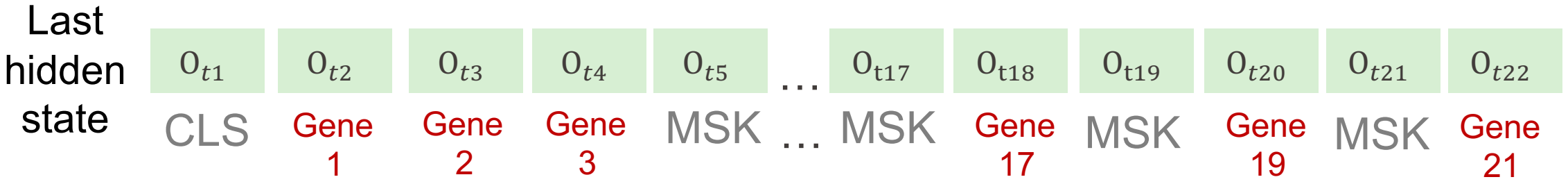
scGPT: Masked Gene Expression

How to leverage cell embeddings in pre-training?

What would be the equivalent of a sentence-level prediction task in scRNA?



scGPT: Gene Generation by Cell Prompt



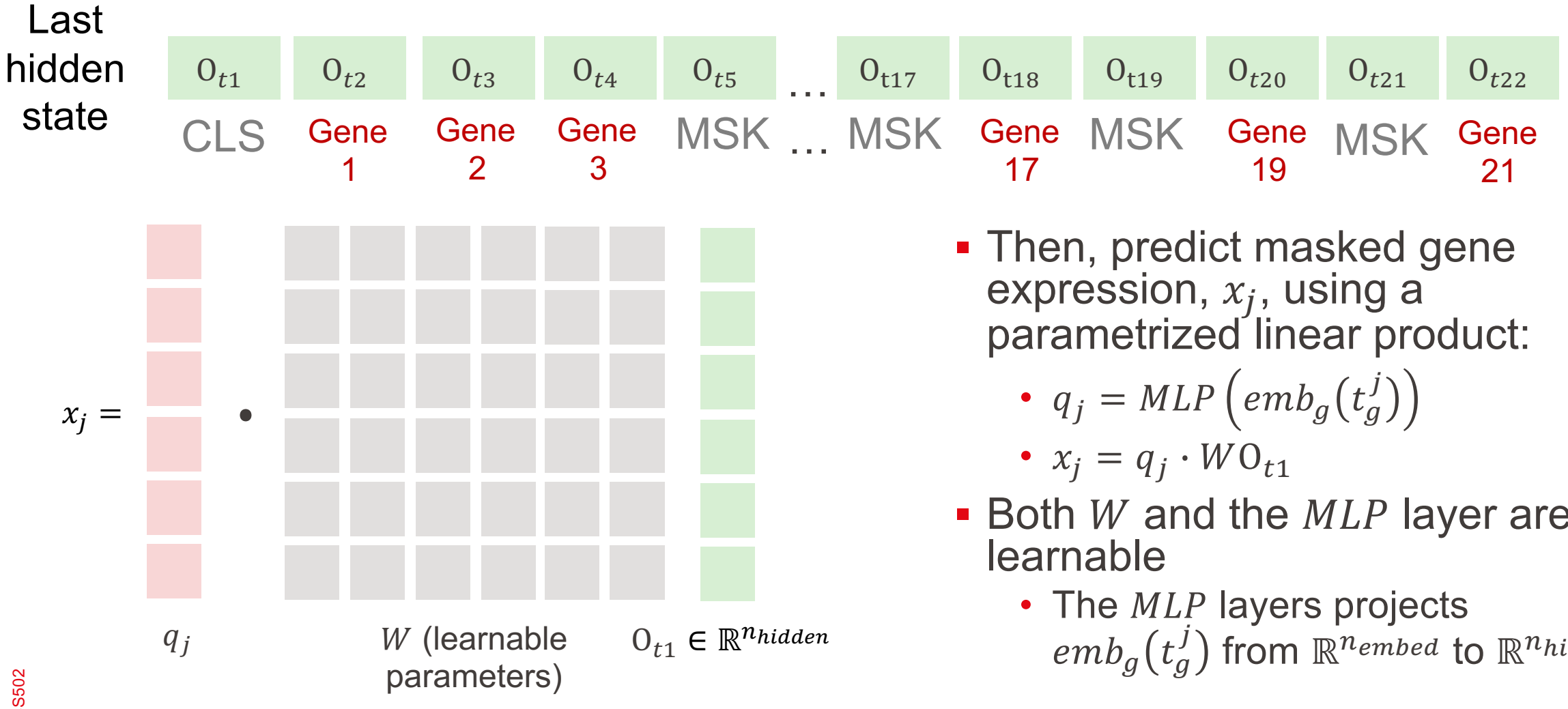
Objective: Predict expression x_j of masked gene j in cell i from the cell representation of O_{t1}

Why is masked gene prediction not enough?

We want to learn useful cell-level representations for downstream tasks!

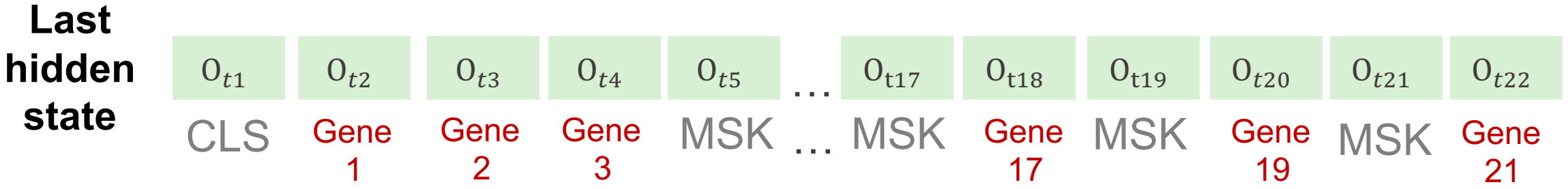
- First, create a query vector, q_j, from the embedded token of gene symbol, t_g^j, e.g. “TP53”, “NOX”
  - $q_j = MLP\left(emb_g\left(t_g^j\right)\right)$

scGPT: Gene Generation by Cell Prompt



- Then, predict masked gene expression, x_j, using a parametrized linear product:
  - $q_j = MLP\left(emb_g\left(t_g^j\right)\right)$
  - $x_j = q_j \cdot W O_{t1}$
- Both W and the MLP layer are learnable
  - The MLP layers projects $emb_g\left(t_g^j\right)$ from $\mathbb{R}^{n_{embed}}$ to $\mathbb{R}^{n_{hidden}}$

scGPT: Putting Both Auxiliary Tasks Together



$$L = \underbrace{\frac{1}{G_{mask}}}_{\text{set of masked genes}} \sum_{j \in G_{mask}} \underbrace{\left(MLP\left(O_{t_{j+1}}\right) - x_j\right)^2}_{\text{masked gene prediction}} + \underbrace{(q_j \cdot W O_{t_1} - x_j)^2}_{\text{gene prediction by cell prompt}}$$

What has scGPT learned?

- Functional relationship between genes by predicting unknown gene expressions from known ones
- High-quality cell representation that can generate expression level of any gene

A Tale of Two Strategies: Encoder vs. Decoder-based Models

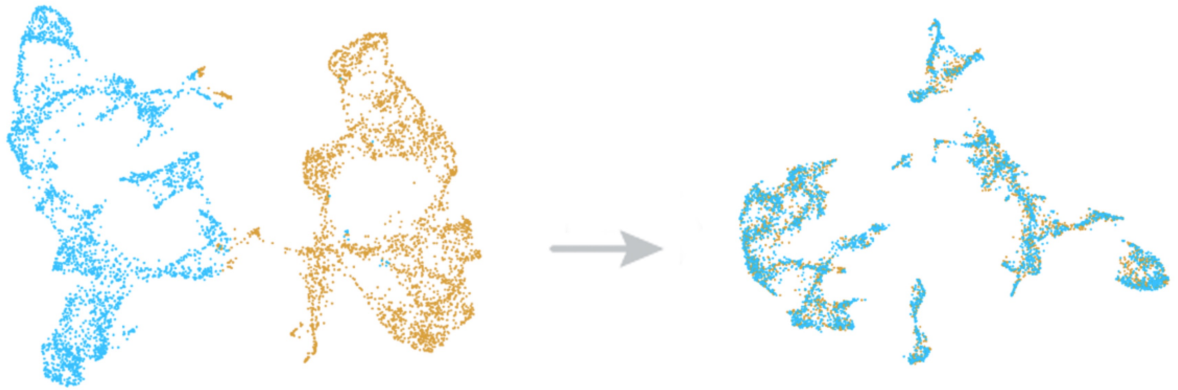
Encoder-based models (e.g. DNABert)

- Bi-directional context
- Fully populated attention matrix

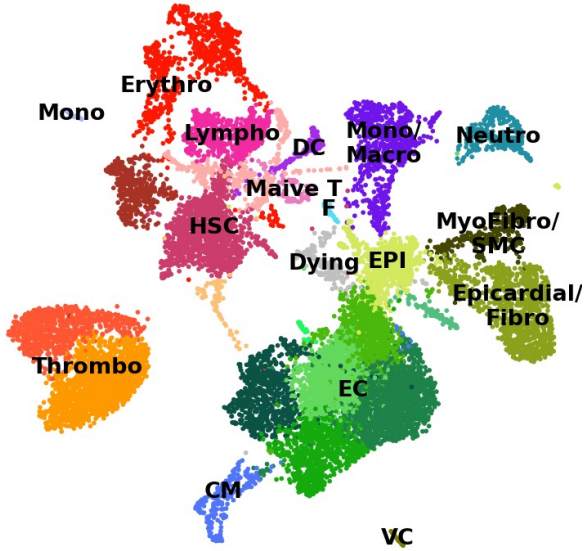
Decoder-based models (e.g. scGPT)

- Autoregressive generation
- Attention masking

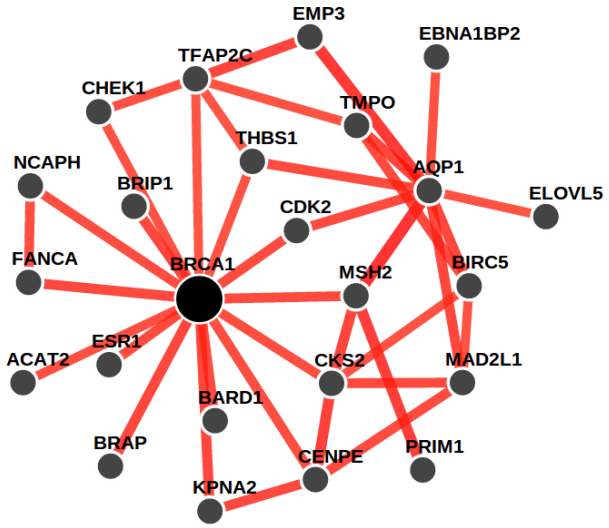
scGPT: Fine-Tuning Tasks



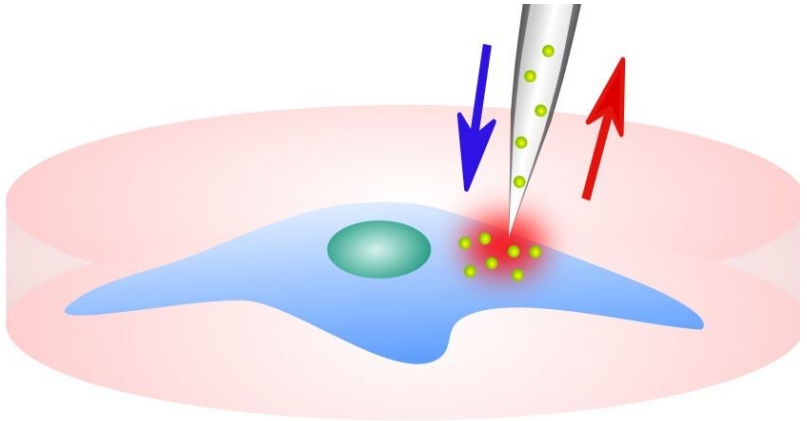
Batch effect correction



Cell type annotation



Gene network inference



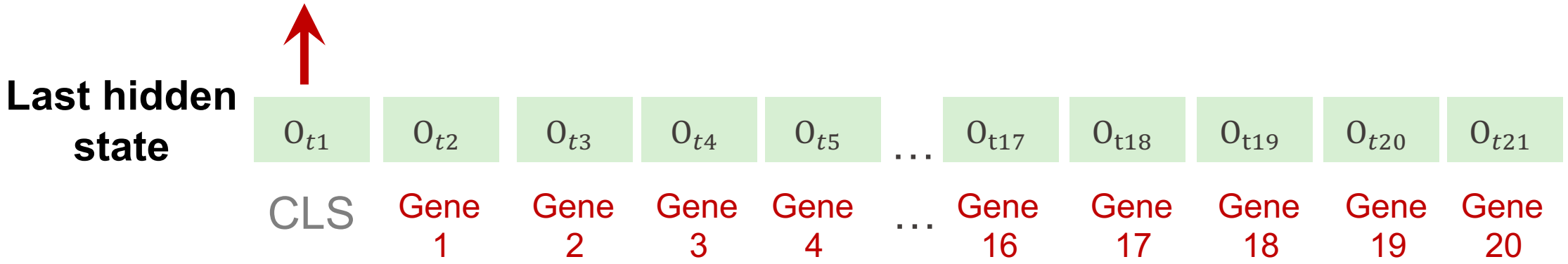
Perturbation response prediction

scGPT: Fine-tuning for Cell Type Annotation

Cell type annotation

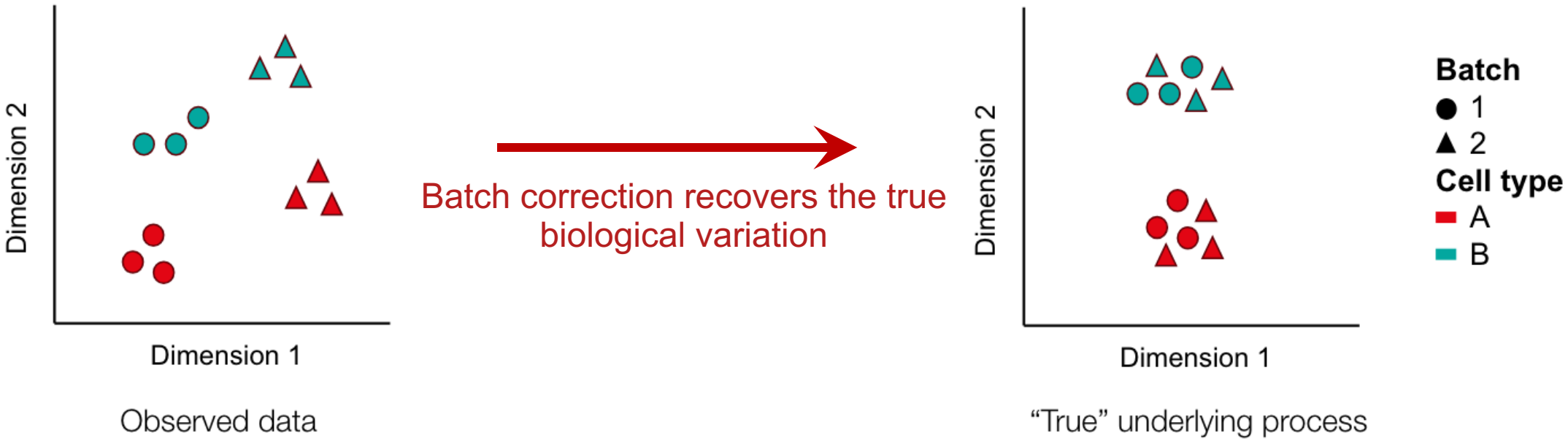
- Use the learned cell embedding, $h_c^{(i)}$ annotate cell i
- Use a MLP layer to predict the annotations: $\hat{y} = MLP(h_c^{(i)})$, where $\hat{y}_j = P(y = y_j)$
- Fine-tune with cross-entropy loss:

$$\mathcal{L}^{(i)} = -\sum_{j}^{K} y_j \log(\hat{y}_j)$$

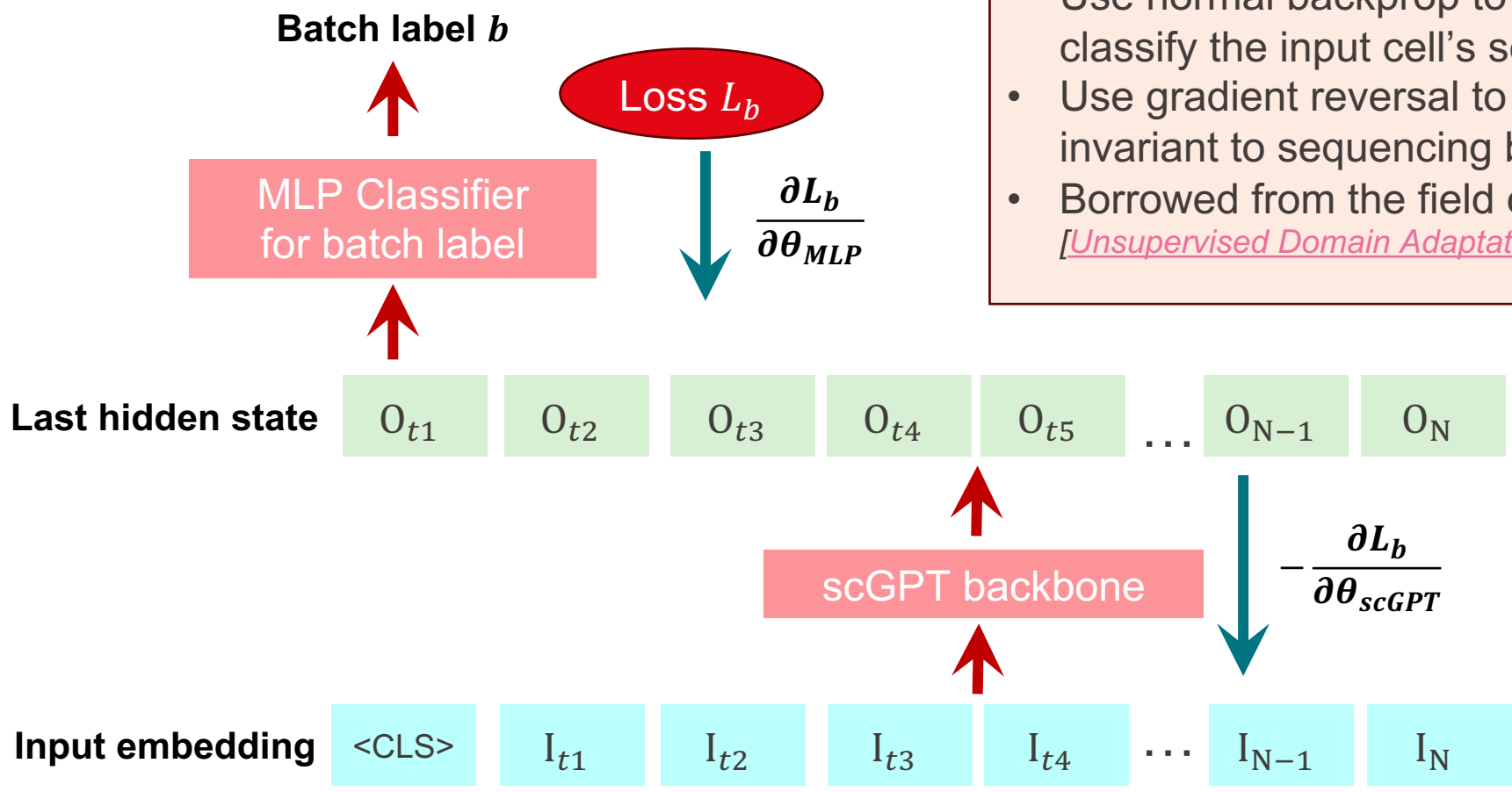


scGPT: Fine-tuning for Batch Correction

- In single-cell data there are significant batch effects due to noise in the technical or experimental processes.



scGPT: Fine-tuning for Batch Correction



Key ideas

- Use normal backprop to train an MLP to classify the input cell's sequencing batch
- Use gradient reversal to learn cell embeddings invariant to sequencing batch
- Borrowed from the field of domain adaptation: *[Unsupervised Domain Adaptation by Backpropagation]*

scGPT: Fine-tuning for Batch Correction

In addition to gradient reversal, facilitate the learning of batch-invariant cell representations, $h_c^{(i)}$, using **elastic cell similarity (ECS)**:

$$L_{ECS} = -(sim(h_c^{(i)}, h_c^{(i')}) - \beta)^2$$

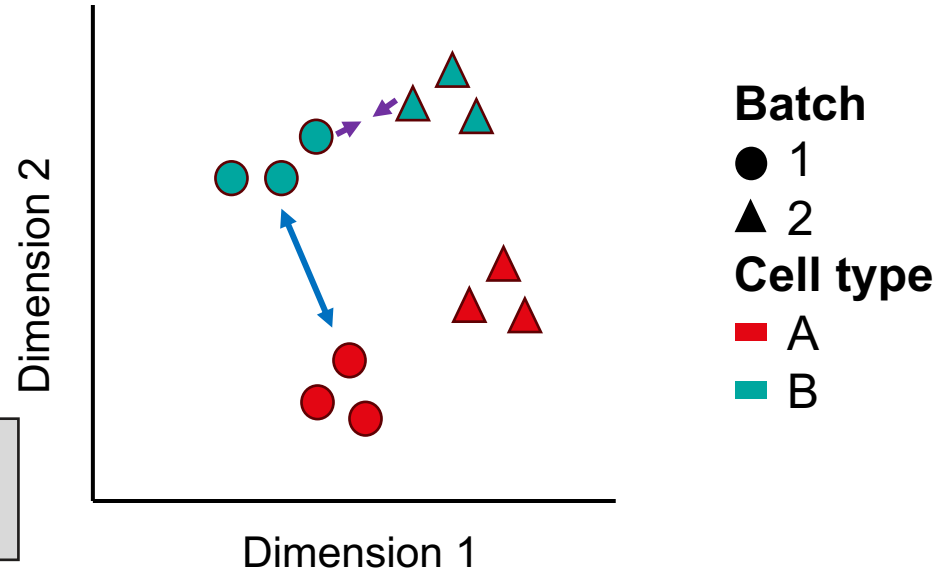
i, i': a pair of cells within the same mini-batch

β: a pre-defined threshold

$sim()$: cosine similarity function

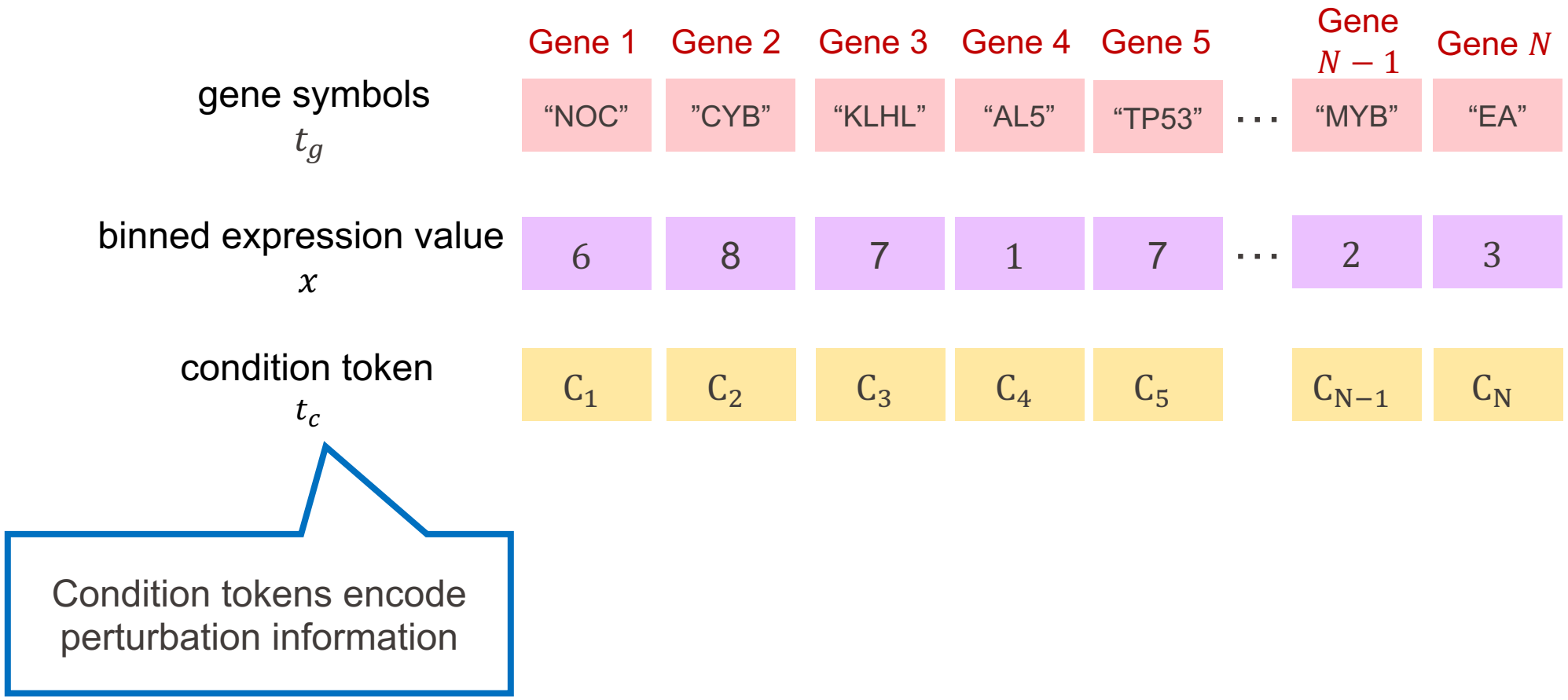
Similar cells (e.g. cells of same type) are pushed closer

Dissimilar cells (e.g. cells of different types) are pushed apart

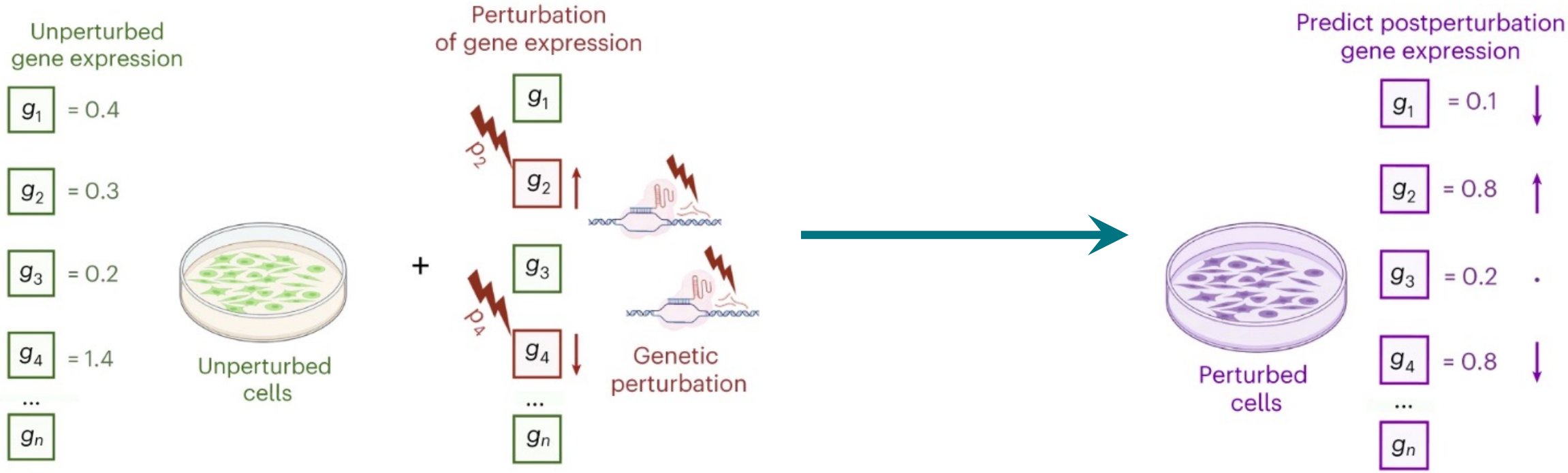


UMAP of Embedding Space

scGPT: Input Embedding



scGPT: Fine-Tuning for Perturbation Response Prediction



Goal: Given a set of observed perturbations, predict the perturbation response of new gene combinations

scGPT: fine-tuning for perturbation response prediction

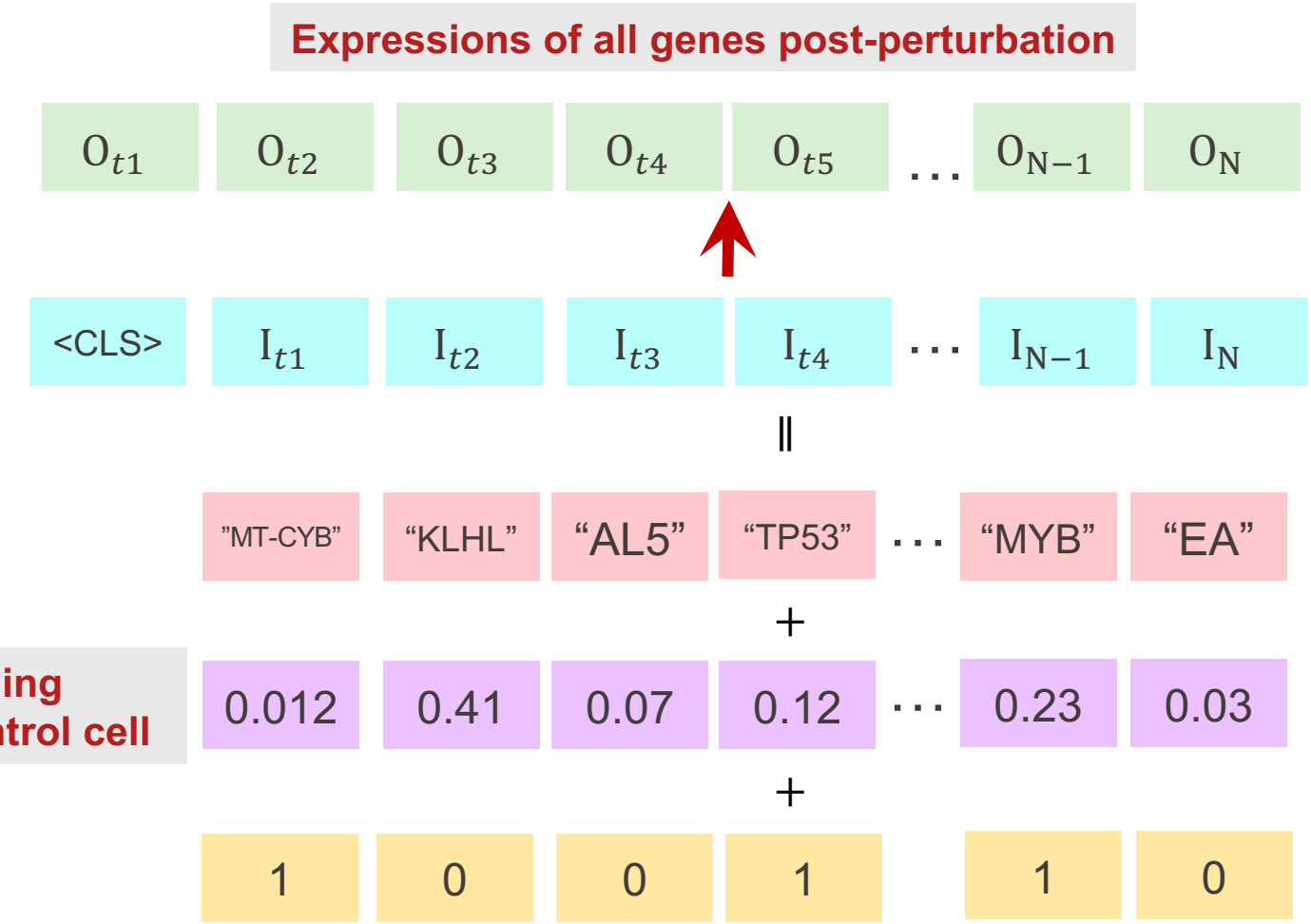
Perturbation response prediction

- Pre-training: predict expression levels of masked genes

$$f(X_{masked}) = \hat{X}$$

- Fine-tuning: predict post-perturbation expression levels of all genes (P parametrizes which genes were perturbed)

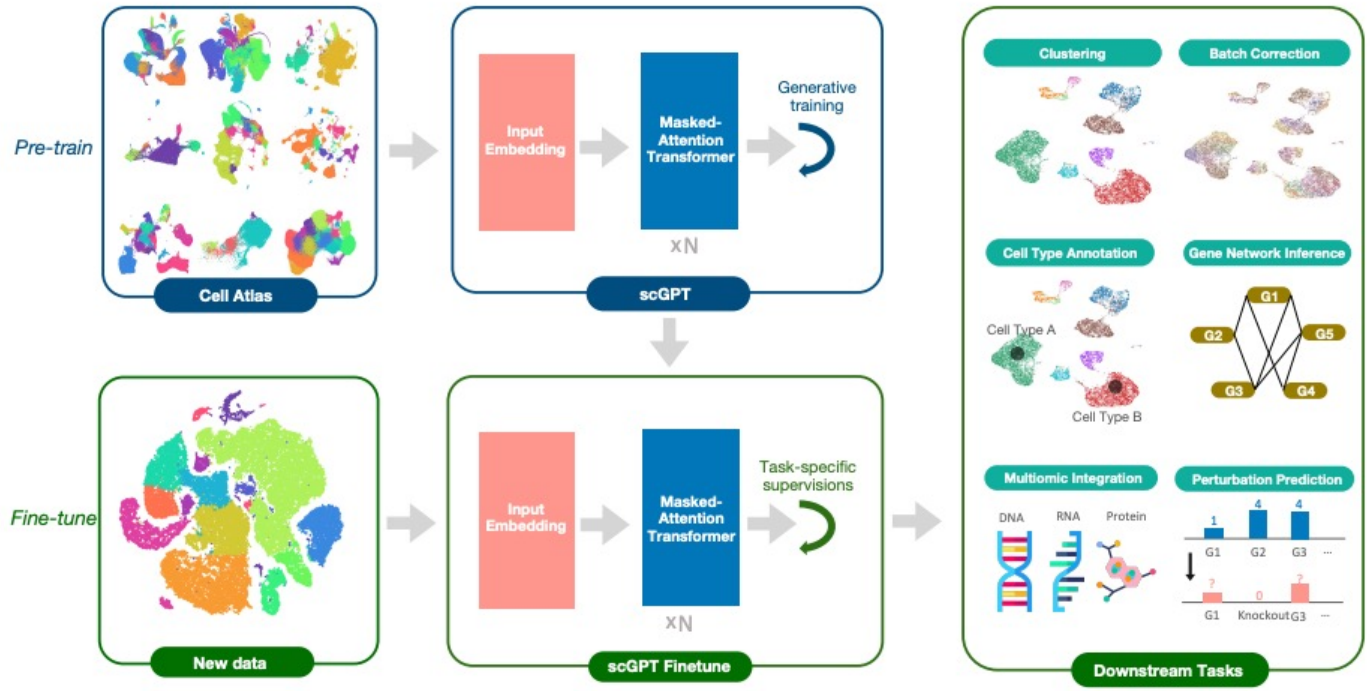
$$f(X_{control}, P) = X_{perturbed}$$



No masked genes and no binning
Input expression levels from a control cell

ScGPT

- **Interested in more details ?**
  - Check the publication: https://www.biorxiv.org/content/10.1101/2023.04.30.538439v2.full.pdf



Recap

- Biological sequence modeling:
  - DNA → RNA → Protein
- DNABERT
  - BERT-like model
  - DNA tokenization based on k-mers
  - Compared to BERT, mask contiguous tokens
- Single-cell biology:
  - Resolution on individual cell level
  - Tasks: batch correction, cell type annotation, trajectory inference, perturbation response prediction
- ScGPT
  - Non-sequential, autoregressive model
  - Pretraining: Masked gene prediction and gene generation by cell prompt

Any Feedback?

Give us feedback on the lecture:

- https://go.epfl.ch/cs502-lecture-6-feedback

Next lecture will be online:

- https://epfl.zoom.us/j/62092835191?pwd=K1VrdFF4TWFFWTRQNXZ6UTBWc29hZz09