

Transformer Encoder Example: scBERT

Yen-Yi Ho

Department of Statistics



scBERT (single-cell BERT) is a deep learning model that applies Transformer architecture (like BERT from NLP) to single-cell RNA sequencing (scRNA-seq) data.

- Lack of curated marker gene lists
- Batch effects
- Annotate based on gene-gene interactions

scBERT: How It Works

- Input: gene expression profiles from single cells
- Genes are encoded into sequences (ordered or ranked by expression)
- Uses self-attention to capture relationships between genes
- Pretrained on large-scale single-cell datasets
- Fine-tuned for downstream biological tasks

Embedding

- Gene embedding (which gene it is)
- Expression embedding (how much it's expressed)

```
>x = gene_emb1(gene_ids) + bin_emb1(bin_ids)
```

representation=gene identity+expression level

gene_emb

- Genes with similar roles \rightarrow similar vectors
- Co-expressed genes \rightarrow closer in embedding space

Expression Embedding

bin_emb1 captures

- Expression intensity (low vs high)
- gene_emb → “TP53”
- bin_emb → “high expression”

Attention Matrix

- Row i : “Where does gene i get information from?”
- Column j : “how much gene j contributes to others”

A simple 3-token example

	GeneA	GeneB	GeneC
GeneA	0.7	0.2	0.1
GeneB	0.1	0.8	0.1
GeneC	0.3	0.3	0.4

- **Row 1:** GeneA receives 0.7 from itself, 0.2 from GeneB, 0.1 from GeneC.
- **Column 2:** GeneB sends 0.2 to GeneA, 0.8 to itself, 0.3 to GeneC.

Row sums to 1.

Attention Matrix

- directional ($i \rightarrow j \neq j \rightarrow i$)
- normalized across rows (each row sums to 1)

So

- Each gene decides how to distribute its attention
- Not how much total attention it receives

Rows = where a gene gets information (receiving),
columns = how much a gene influences others (sending).

Final Takeaway

- Attention = information routing system
- Rows = who I listen to
- Columns = who listens to me
- Output = weighted mixture of others