

Single Dell Data Analysis Course

Dimensionality reduction 1: PCA

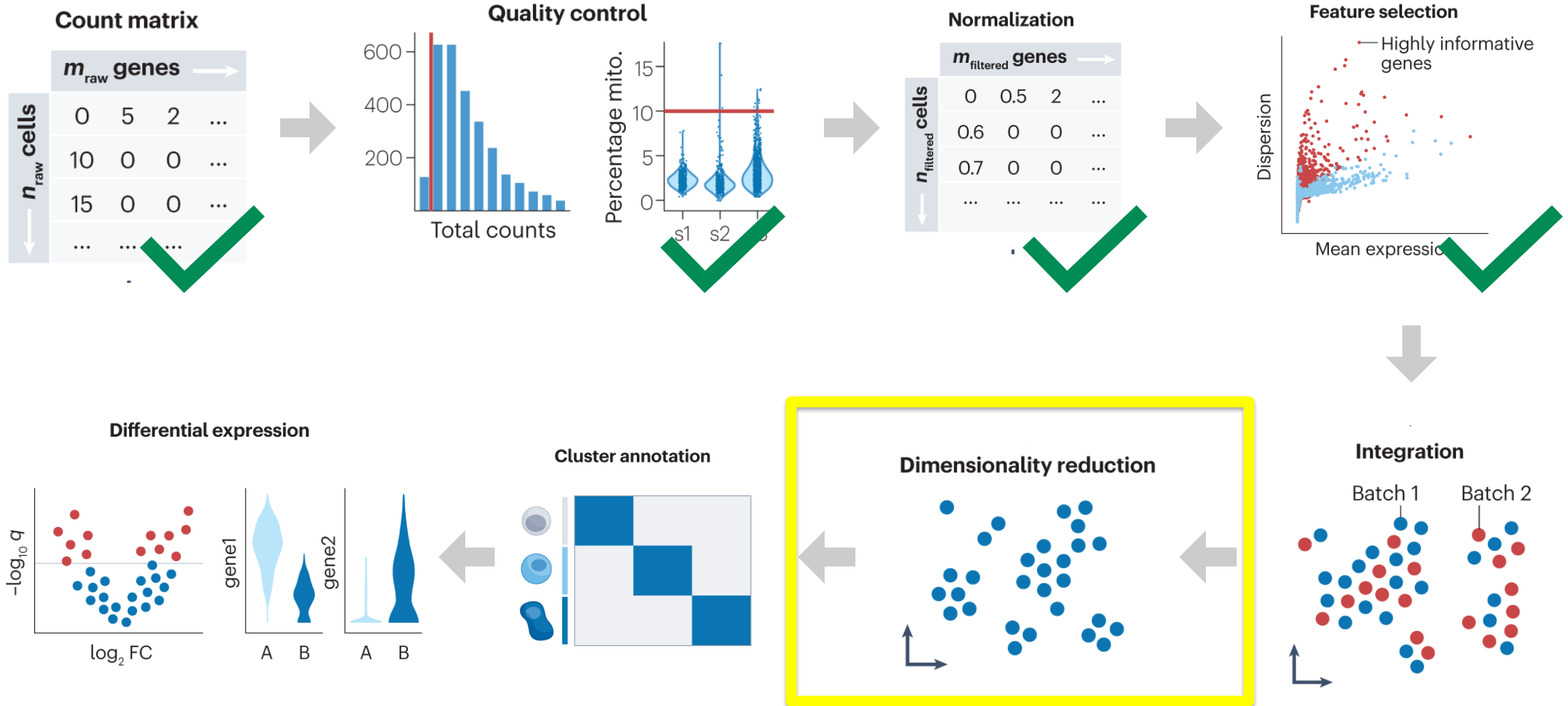
Lisa Buchauer

Professor of Systems Biology of Infectious Diseases

Department of Infectious Diseases and Intensive Care

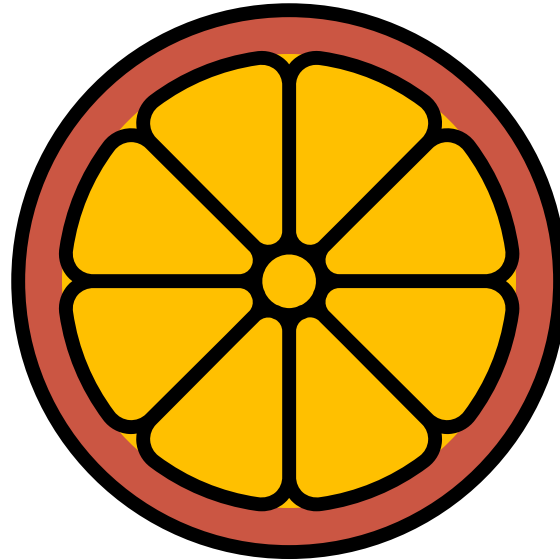
Charité - Universitätsmedizin Berlin

Today



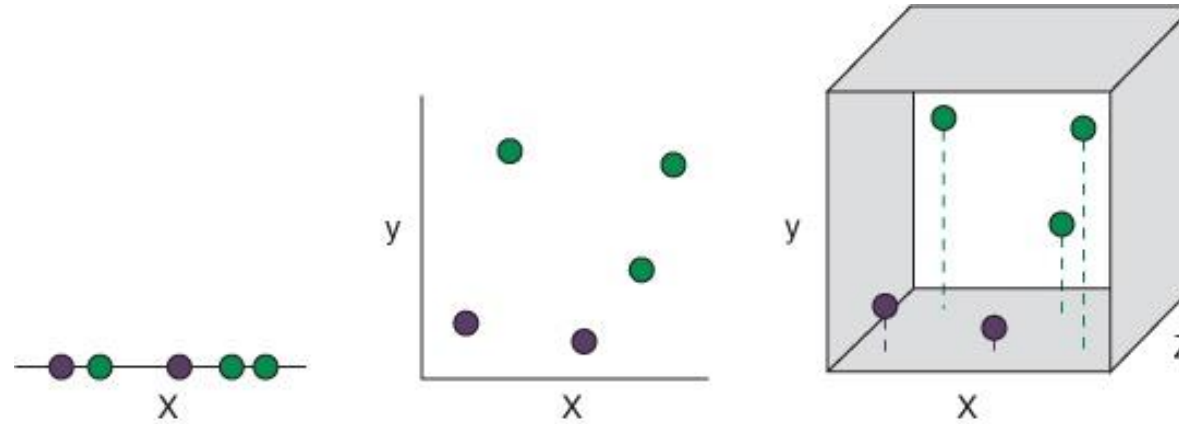
Heumos, L., Schaar, A.C., Lance, C. et al. Best practices for single-cell analysis across modalities. Nat Rev Genet 24, 550–572 (2023). <https://doi.org/10.1038/s41576-023-00586-w>

Losing intuition in high dimensions: a “fun fact”



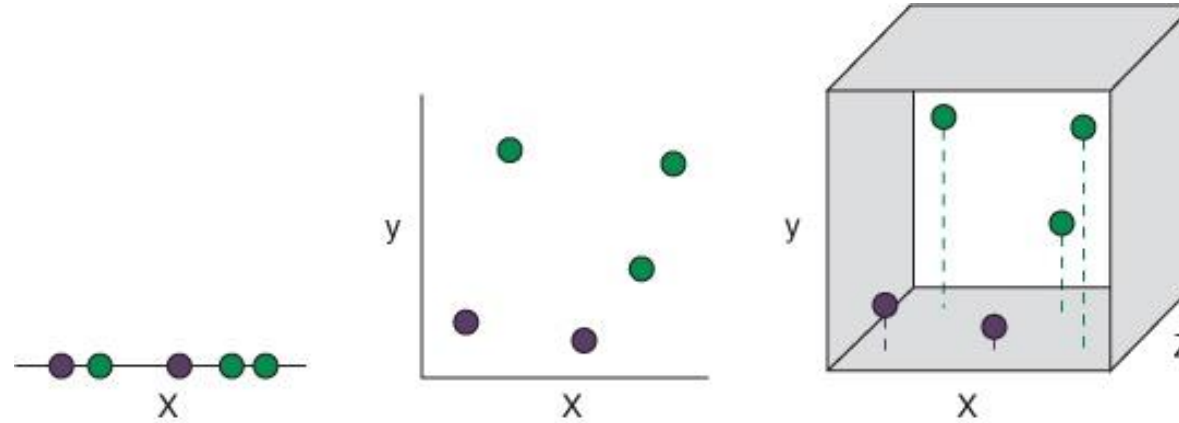
In a high-dimensional orange, most of the mass is in the skin, not in the pulp

“The curse of dimensionality”



data sparsity
most of the high-dim
space is empty

“The curse of dimensionality”



data sparsity
most of the high-dim
space is empty

**distances may lose
meaning**

**computational cost/
algorithmic efficiency**

noise dimensions

**visualization
challenges**

...

Reducing dimensionality



Step 1: Feature selection

scanpy

```
sc.pp.highly_variable_genes(adata, n_top_genes=2000, batch_key="sample")
```

Seurat

```
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)
```

From PCA to clusters and UMAP in basic tutorials



1

```
sc.tl.pca(adata)
```

2

```
sc.pp.neighbors(adata)
```

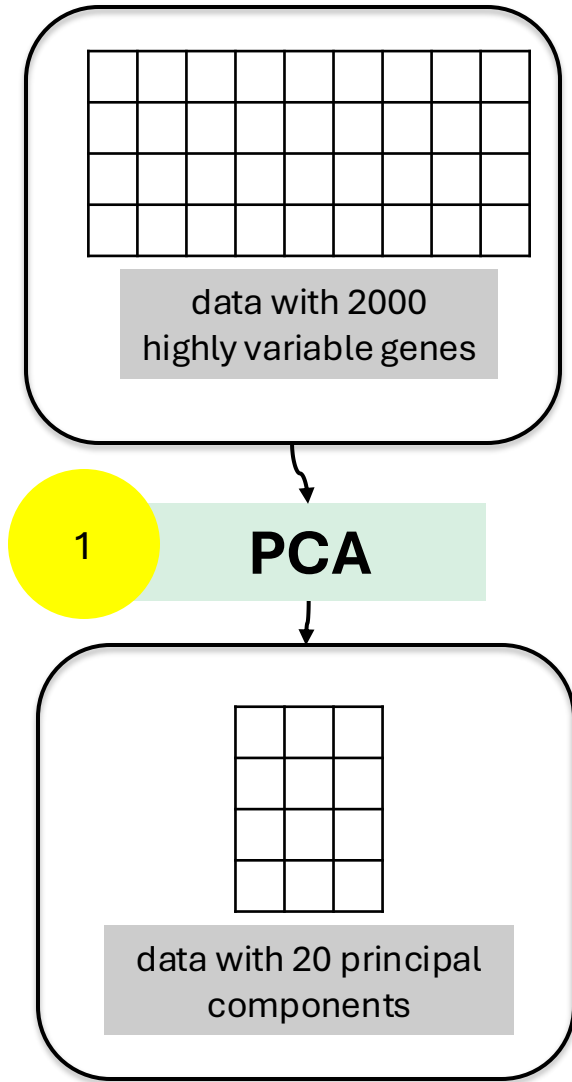
3

```
# Using the igraph implementation and a fixed number of iterations can be significantly faster  
sc.tl.leiden(adata, flavor="igraph", n_iterations=2)
```

4

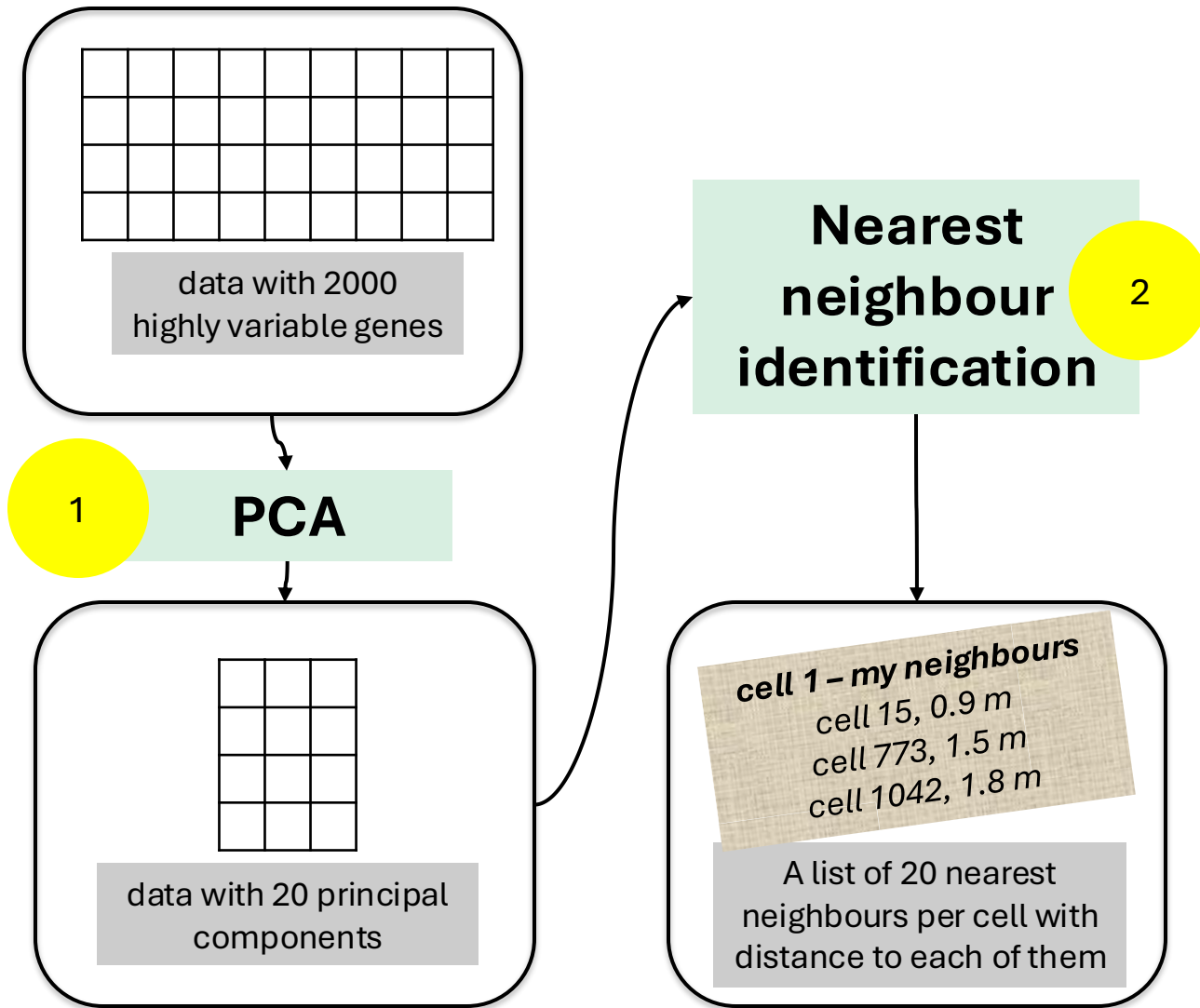
```
sc.tl.umap(adata)
```

Data types along the processing path



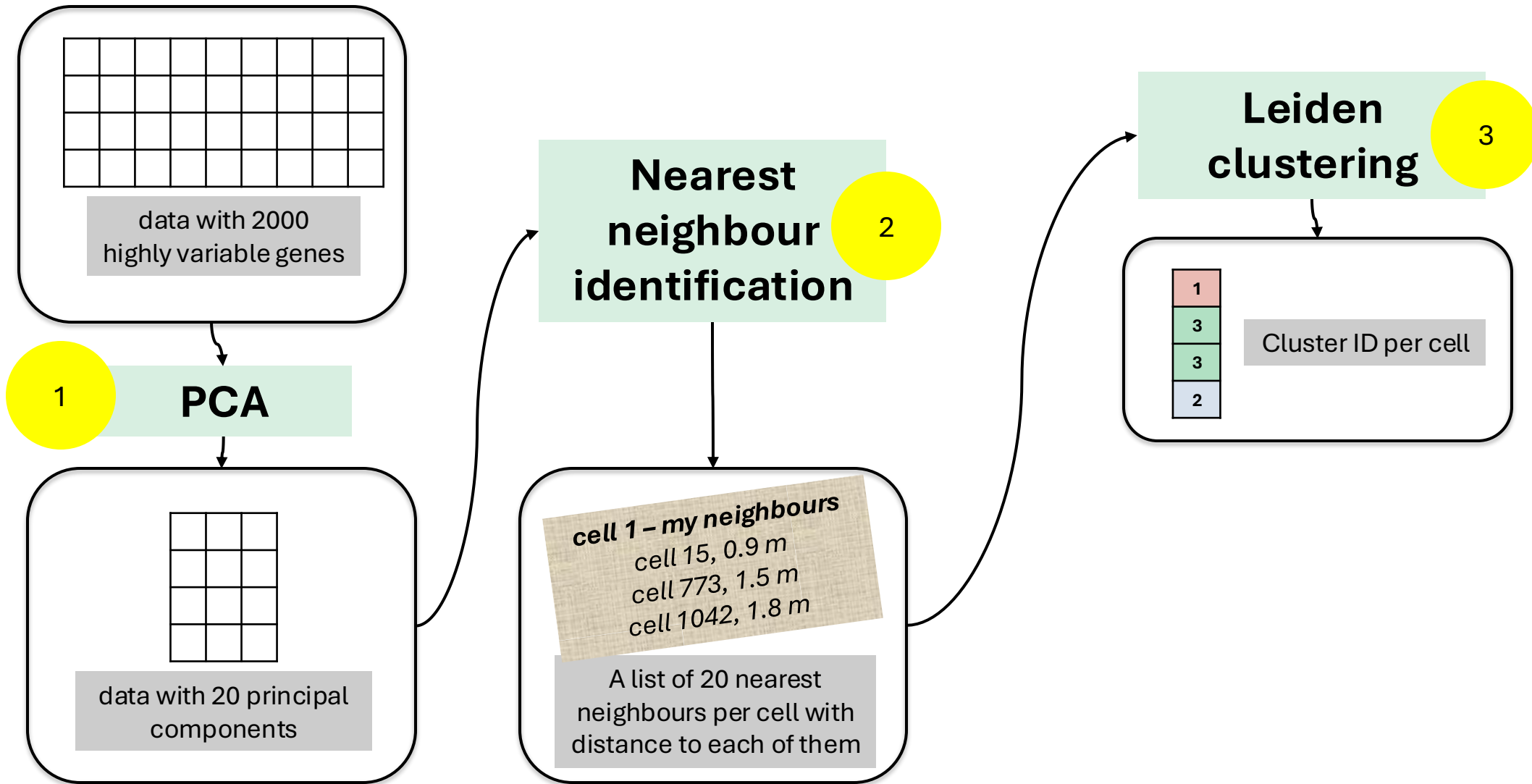


Data types along the processing path



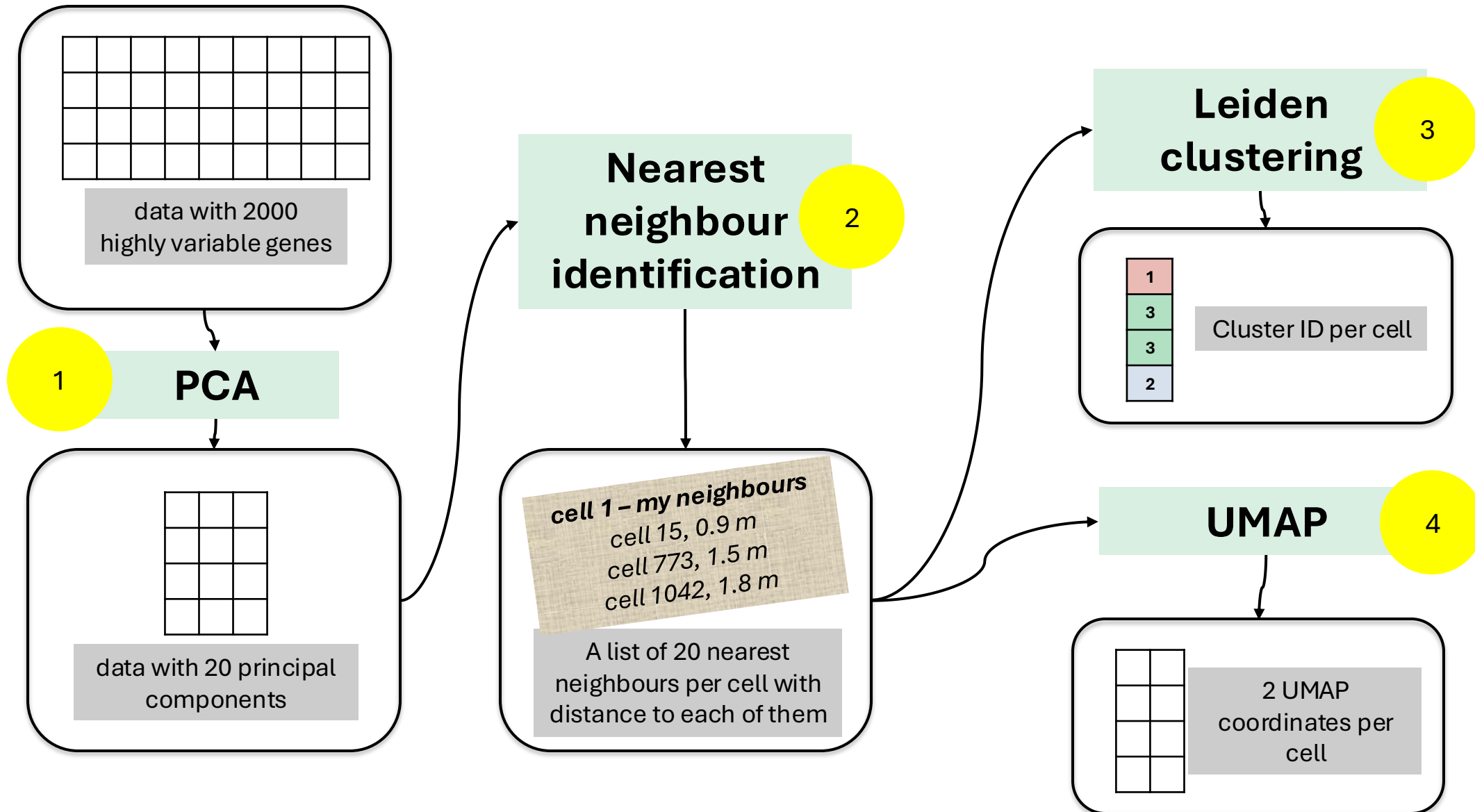


Data types along the processing path

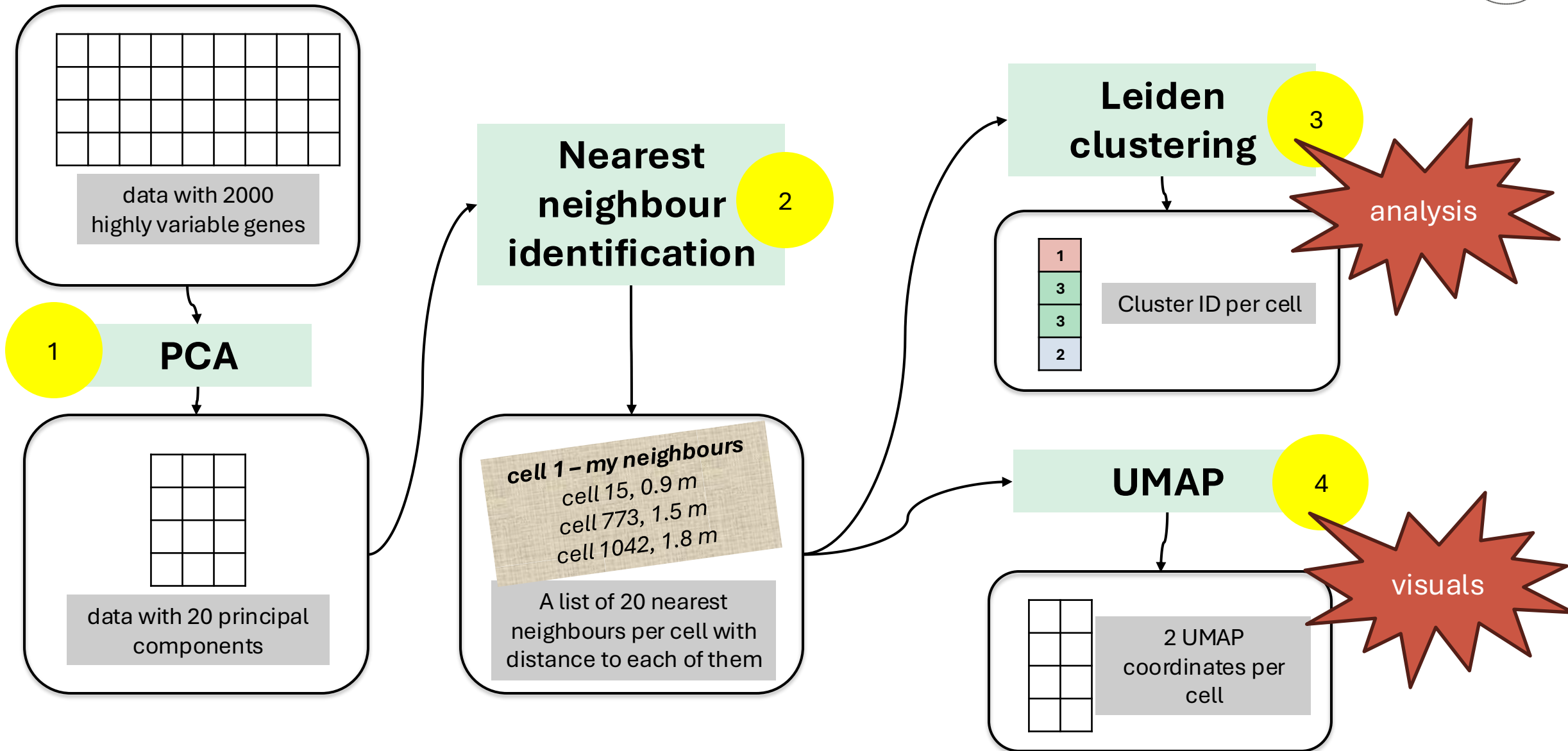




Data types along the processing path



Data types along the processing path



Principal Component Analysis



python

```
sc.tl.pca(adata)
```

R

```
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))
```



Principal Component Analysis

python

```
sc.tl.pca(adata)
```

R

```
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))
```

Motivation

Less dimensions are better for downstream performance and interpretability

Many genes have correlated behaviour – not every original dimension adds value



Principal Component Analysis

python

```
sc.tl.pca(adata)
```

R

```
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))
```

Motivation

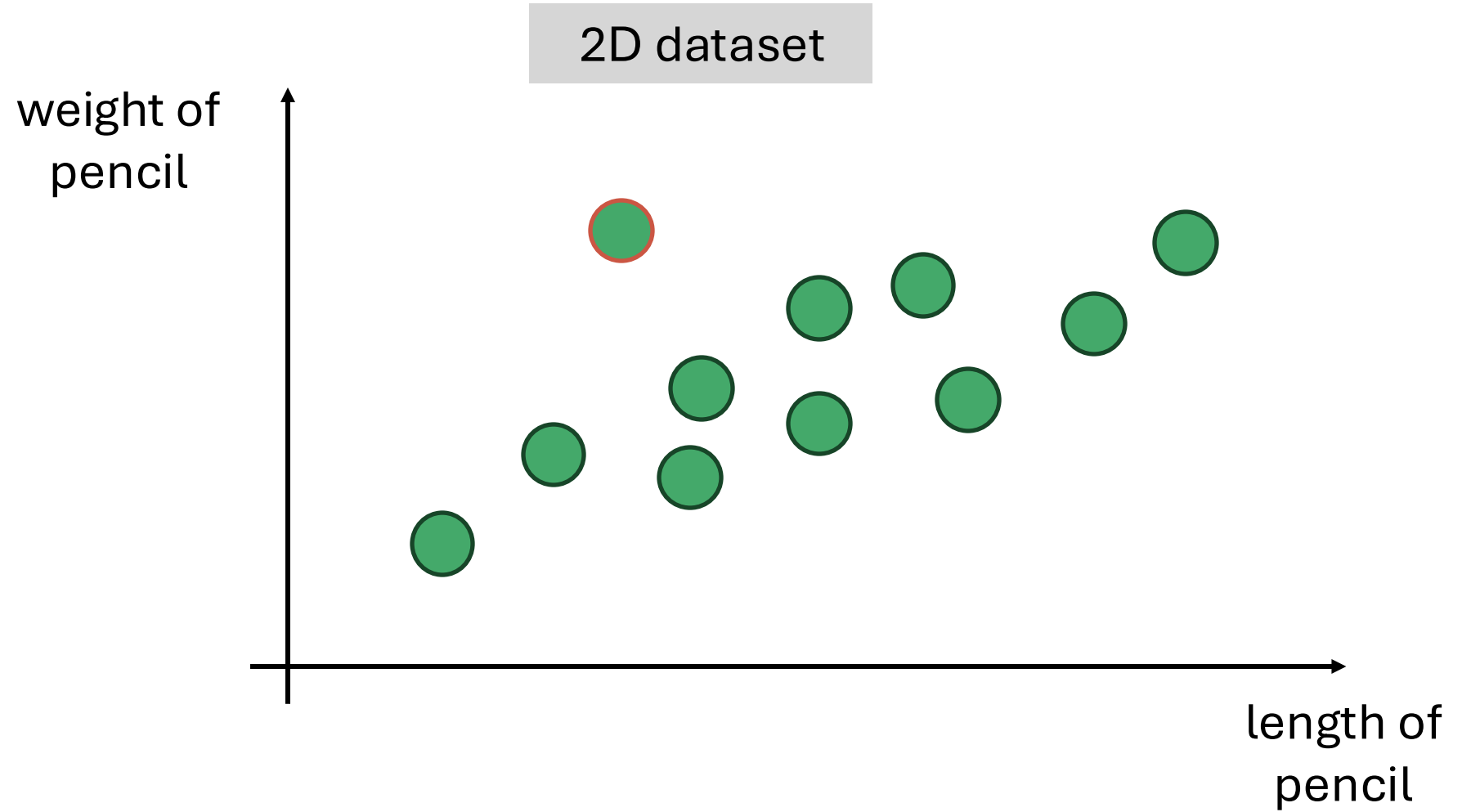
Less dimensions are better for downstream performance and interpretability

Many genes have correlated behaviour – not every original dimension adds value

Goal

Move as much information as possible into as few dimensions as possible

Principal Component Analysis



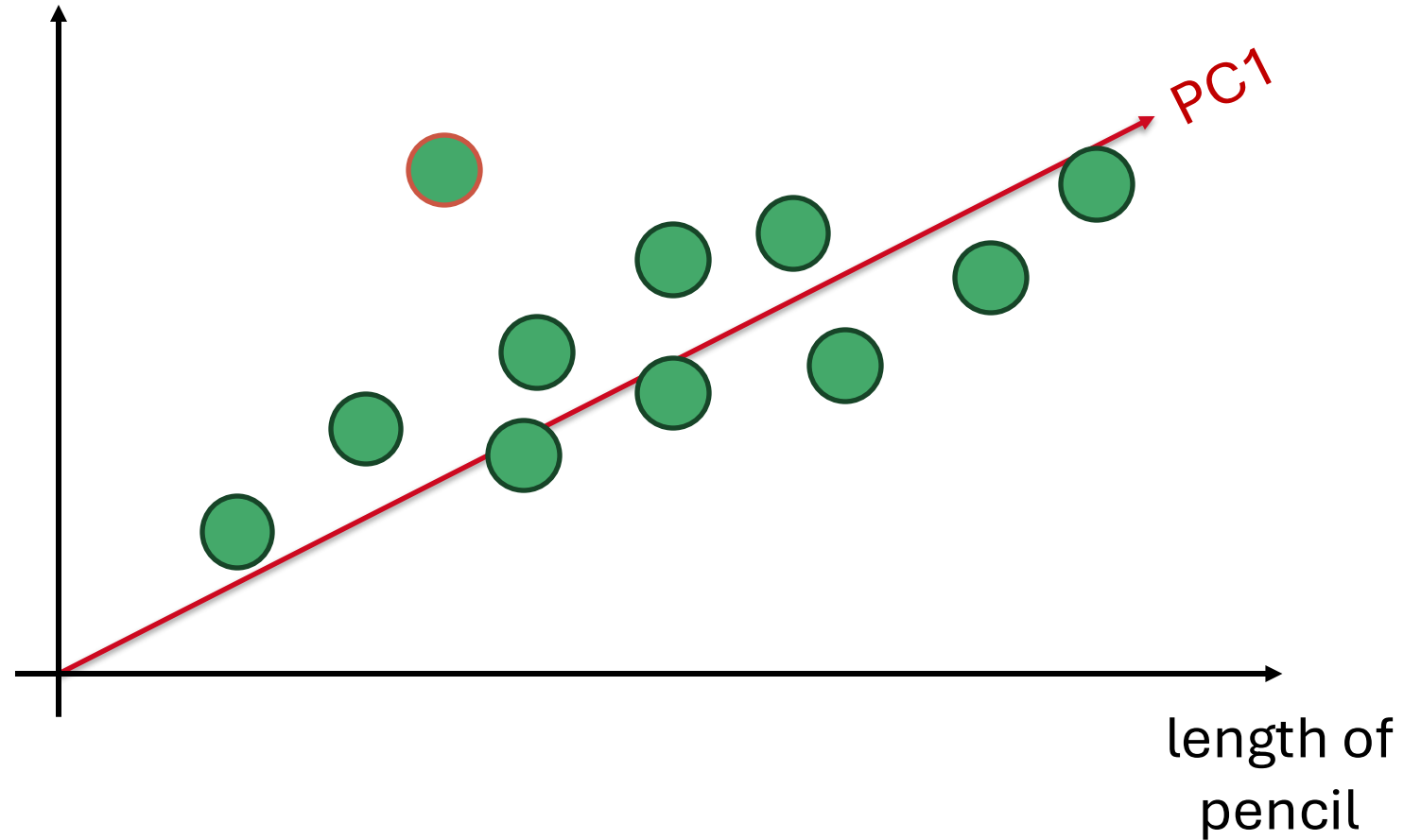
Principal Component Analysis



2D dataset

weight of
pencil

Find the
dimension
which captures
the most
variability



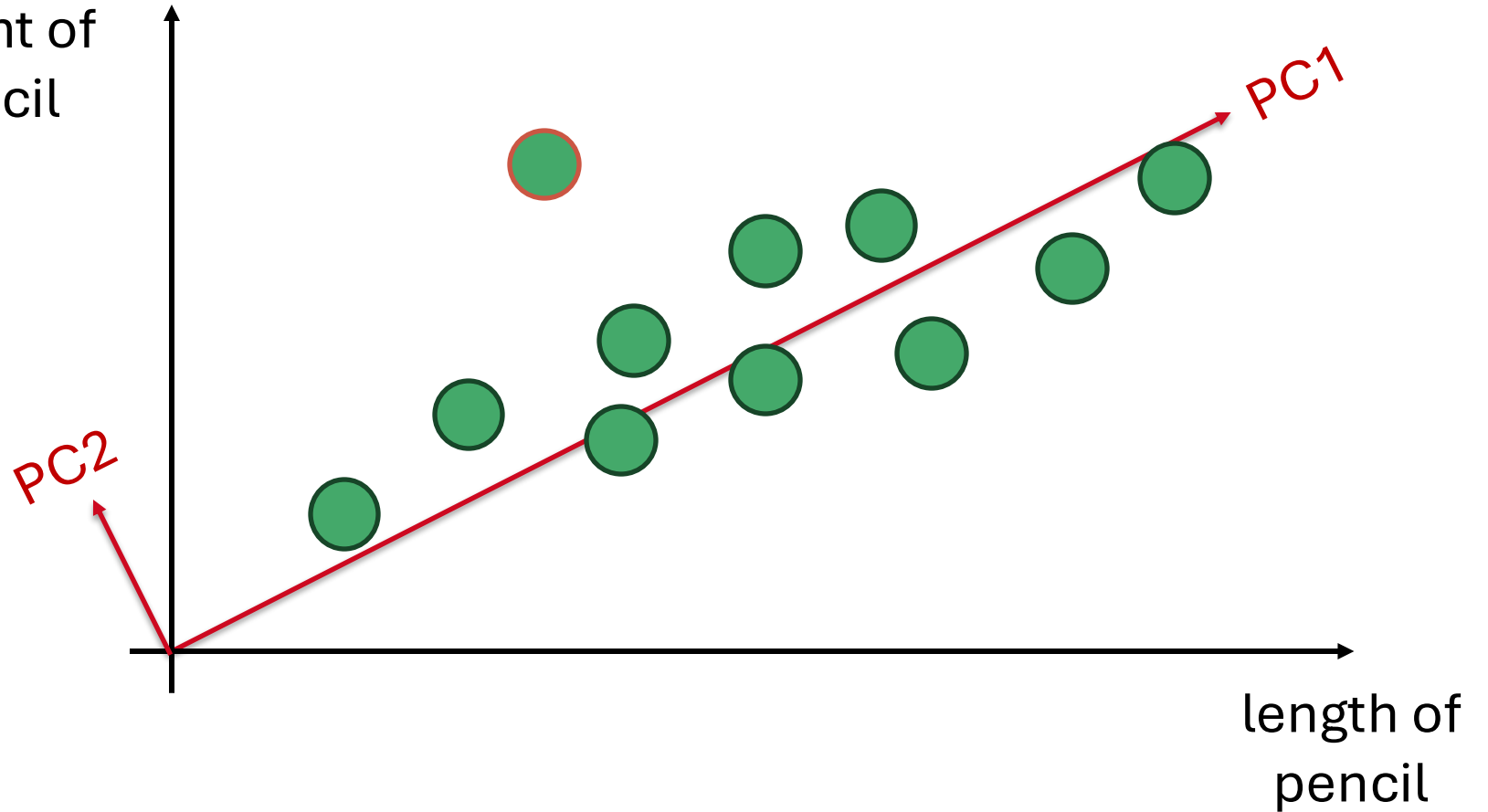
Principal Component Analysis



2D dataset

weight of
pencil

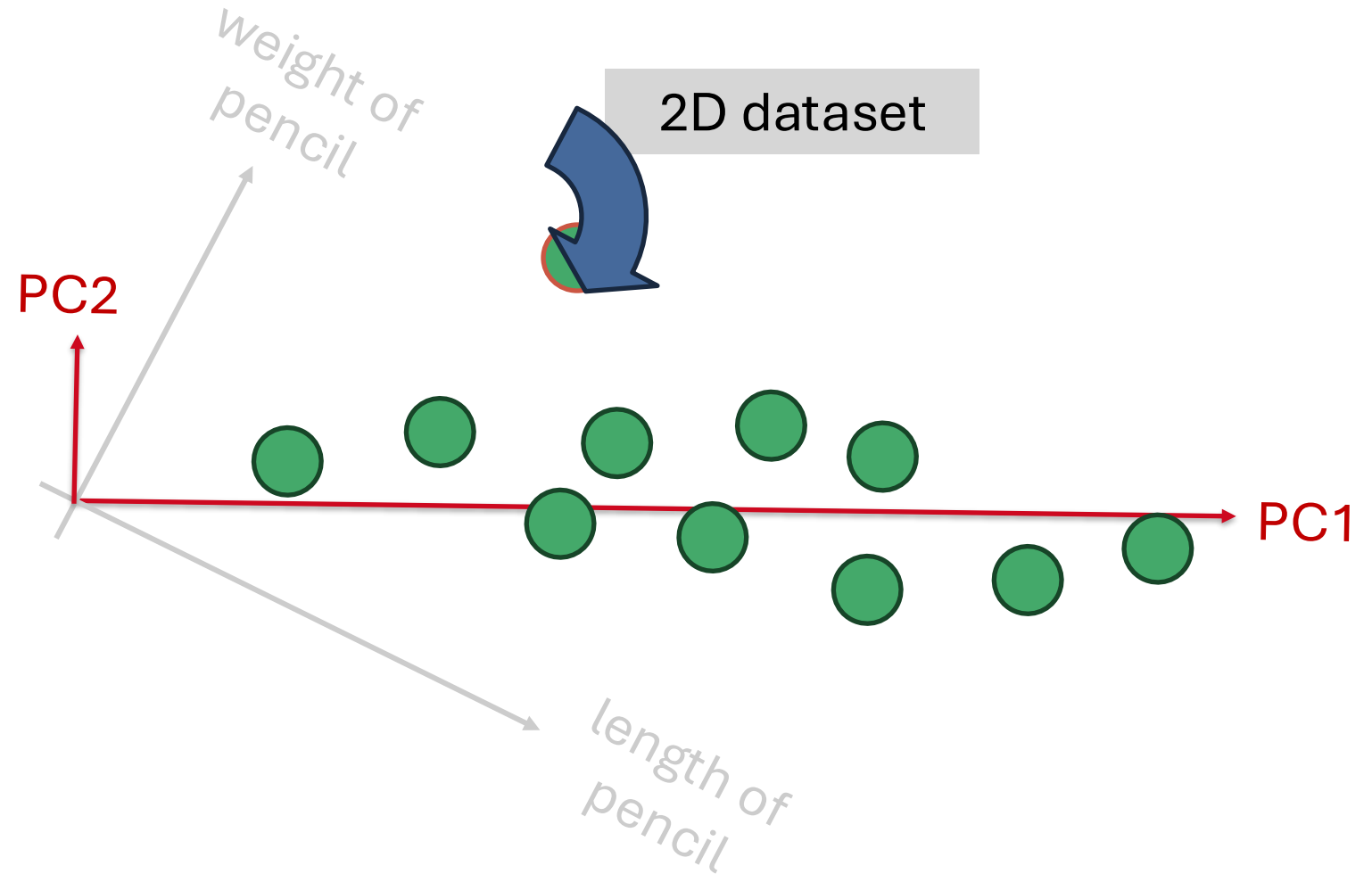
Find the
dimension
which captures
the **2nd** most
variability
(etc)





Principal Component Analysis

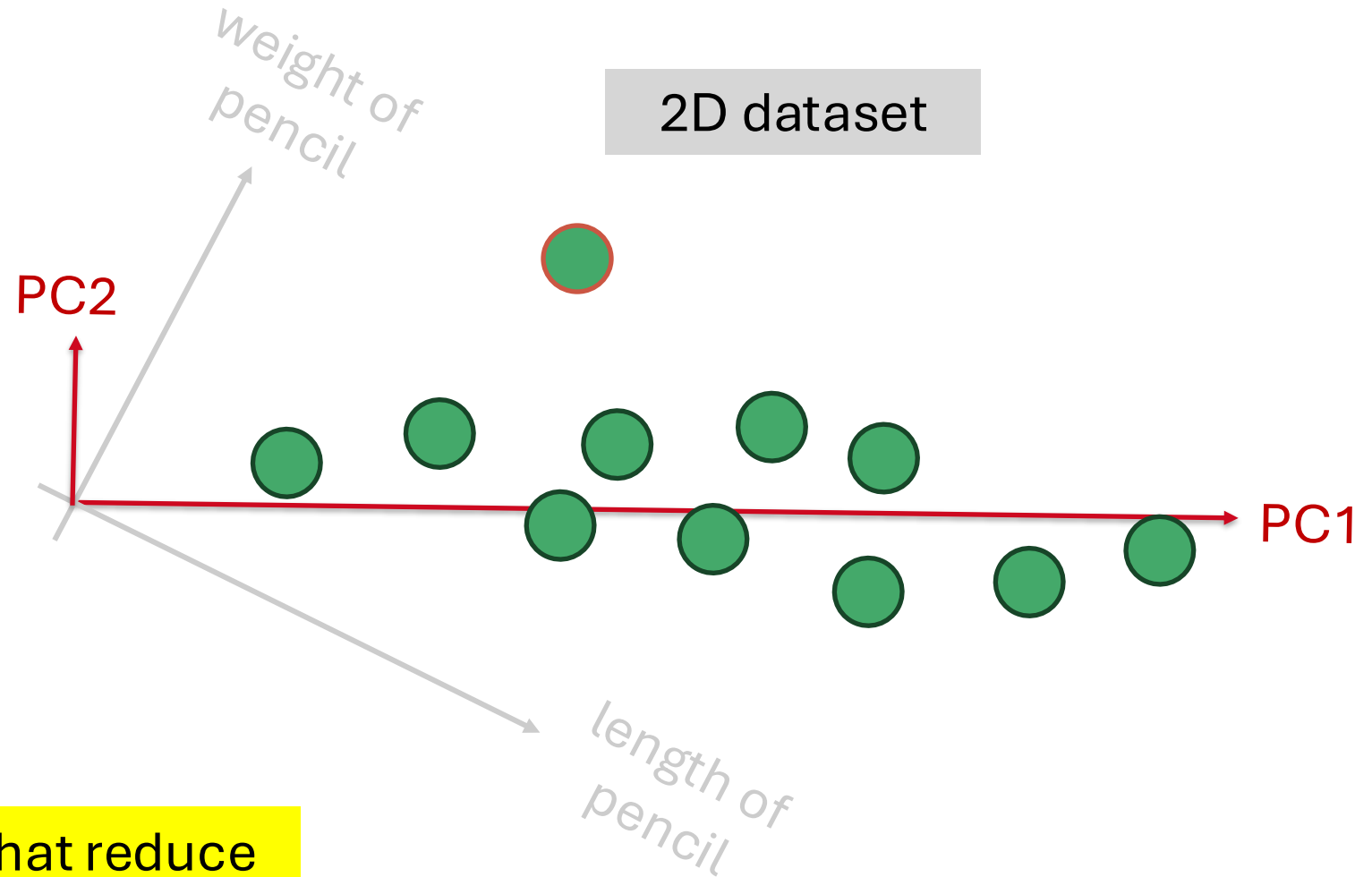
Describe each point in terms of the new rotated coordinates





Principal Component Analysis

2D dataset



Describe each point in terms of the new rotated coordinates

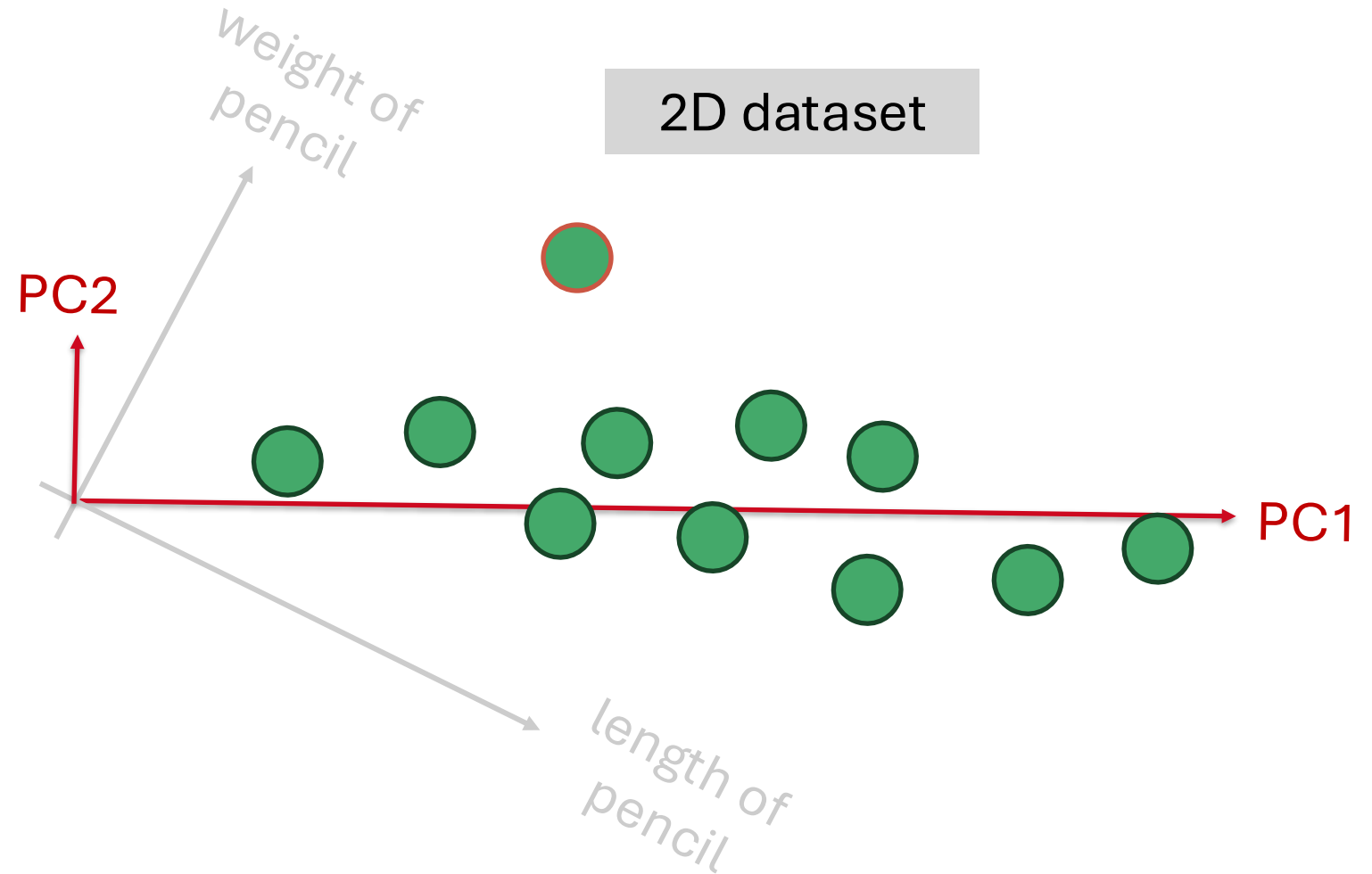
?! How does that reduce dimensionality?

Principal Component Analysis



Inspect how much of the total variance is explained by each PC

	Explained variance
PC1	95%
PC2	5%



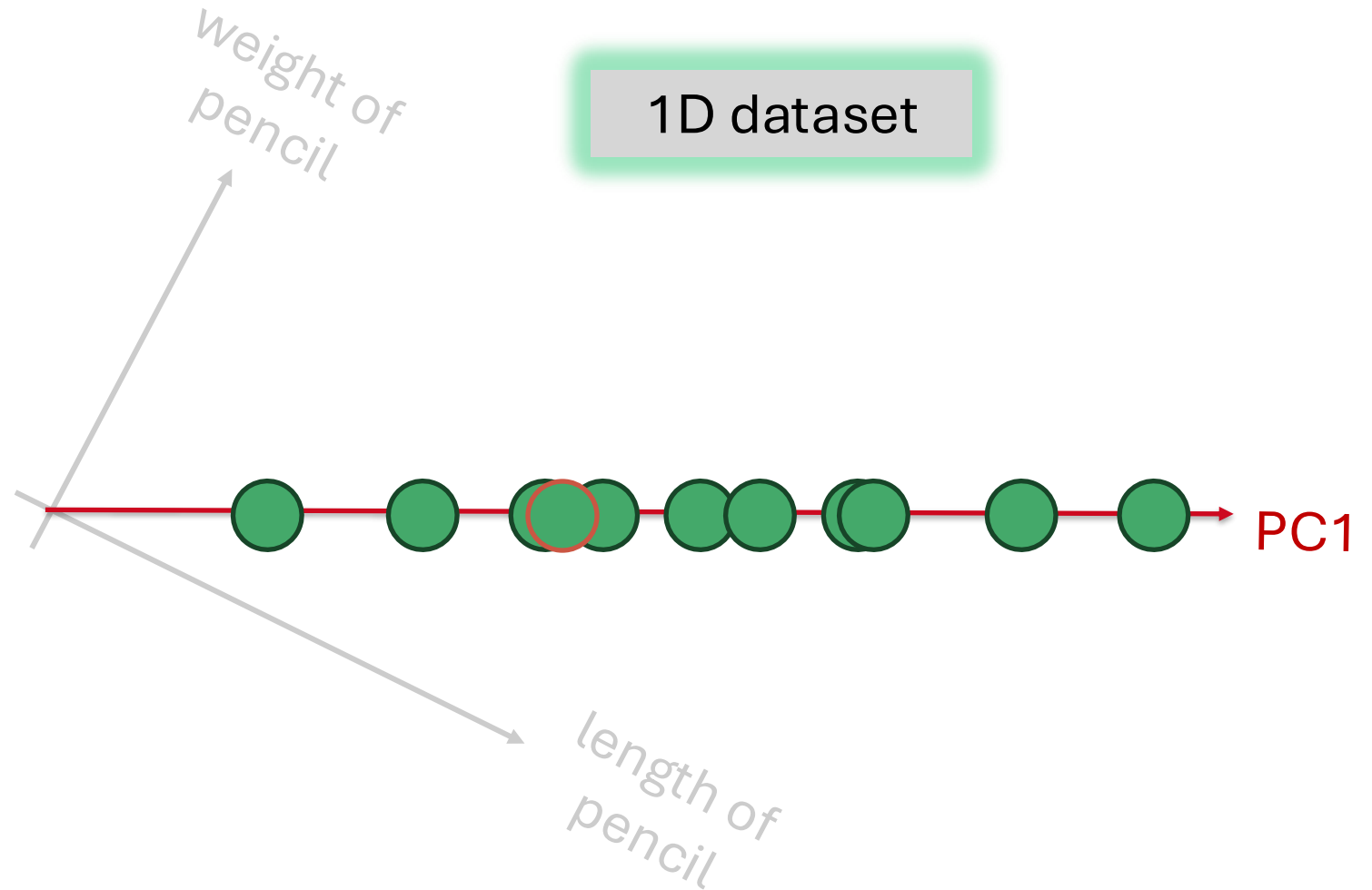


Principal Component Analysis

1D dataset

Discard lesser
important
dimensions

	Explained variance
PC1	95%

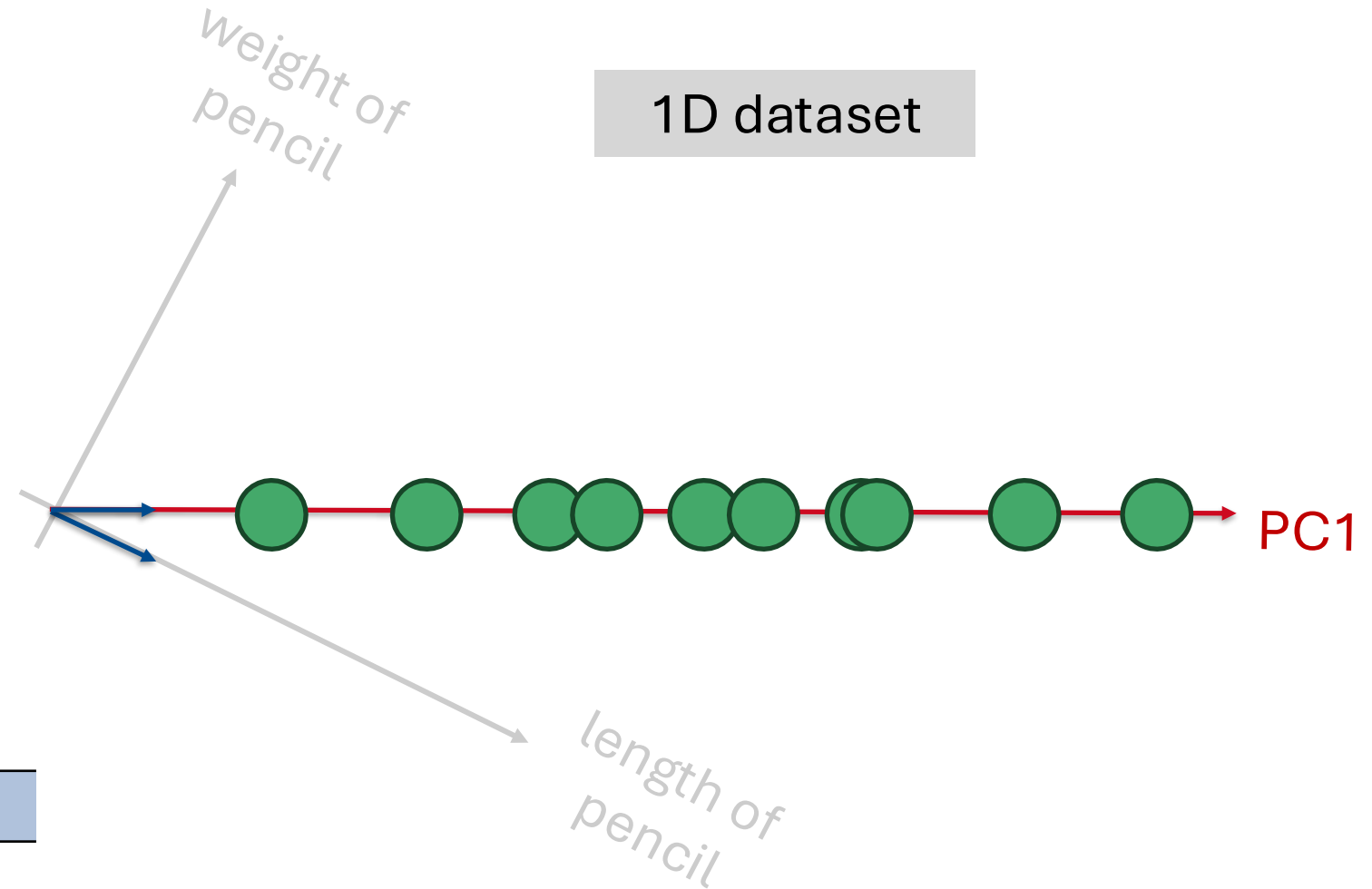




Principal Component Analysis

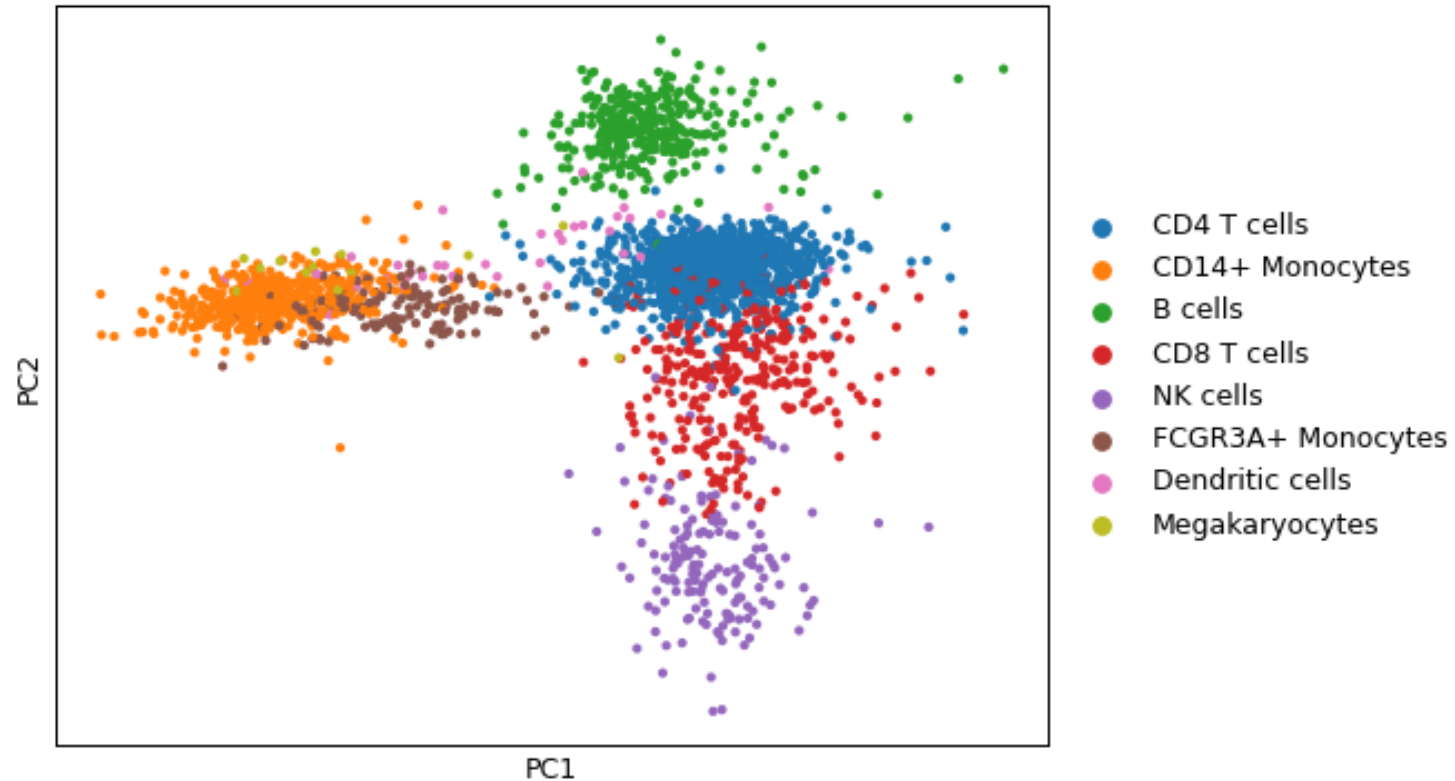
Loadings
(projections of
features onto
PCs) help
interpretability

	Loading on PC1
Length	0.9
Weight	0.2



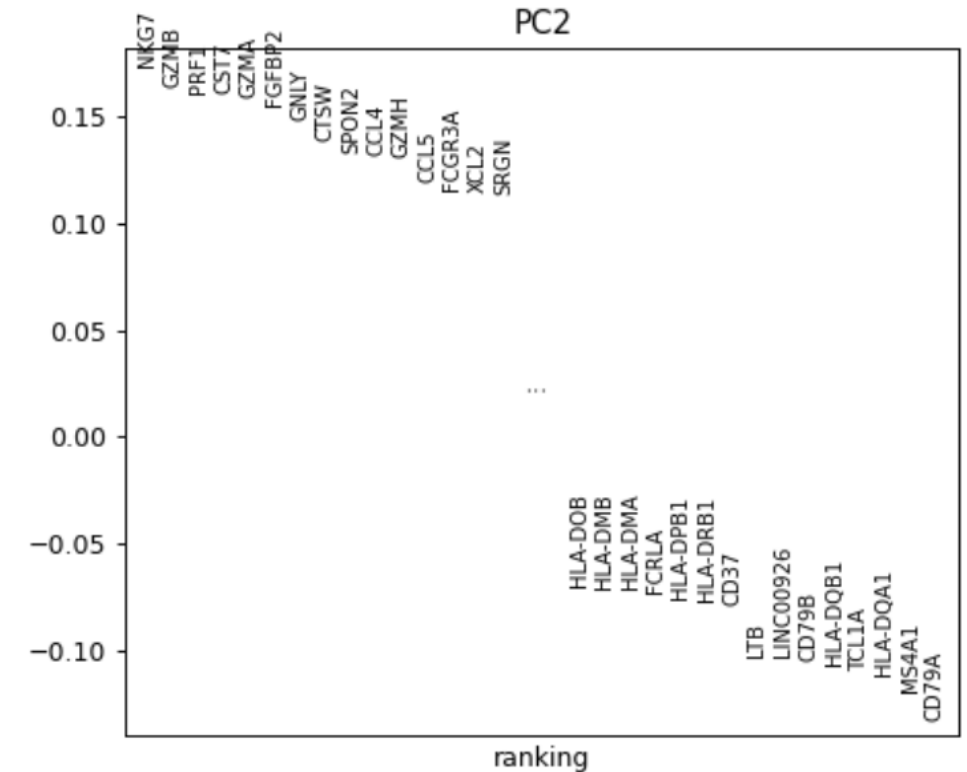
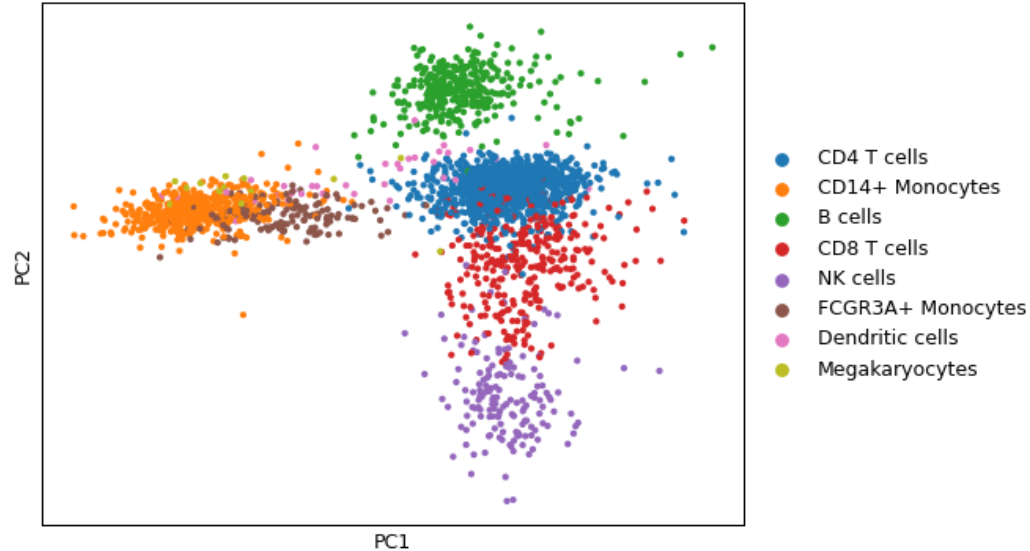


Principal Component Analysis for single cell data



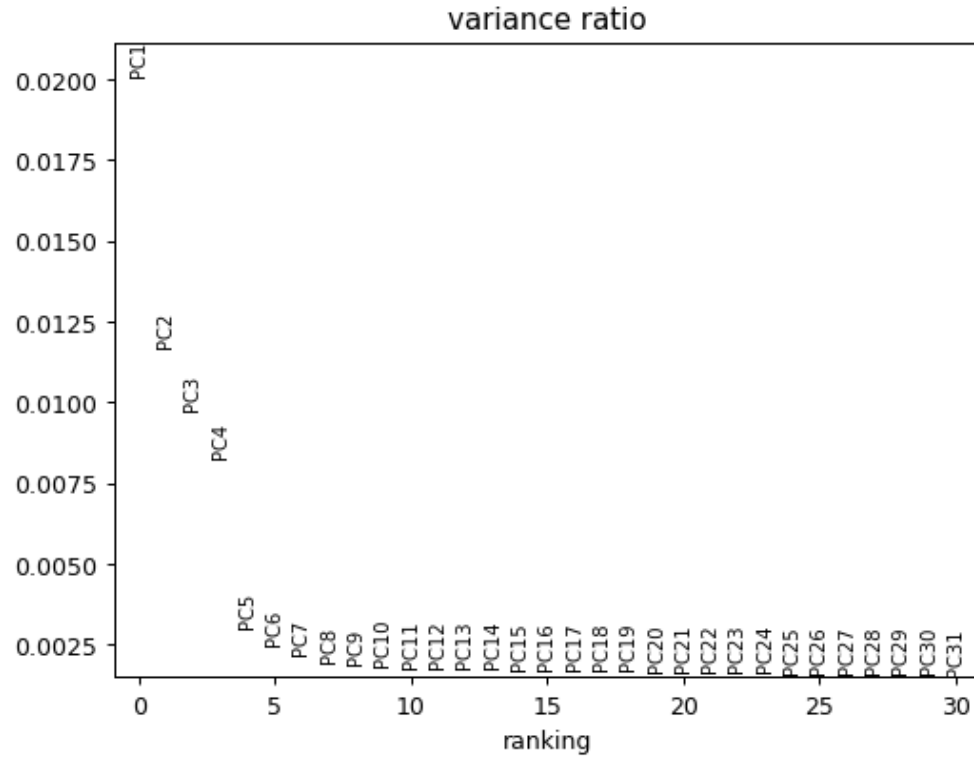
First 2 components are sufficient to separate B cells, T cells and monocytes!

Principal Component Analysis for single cell data



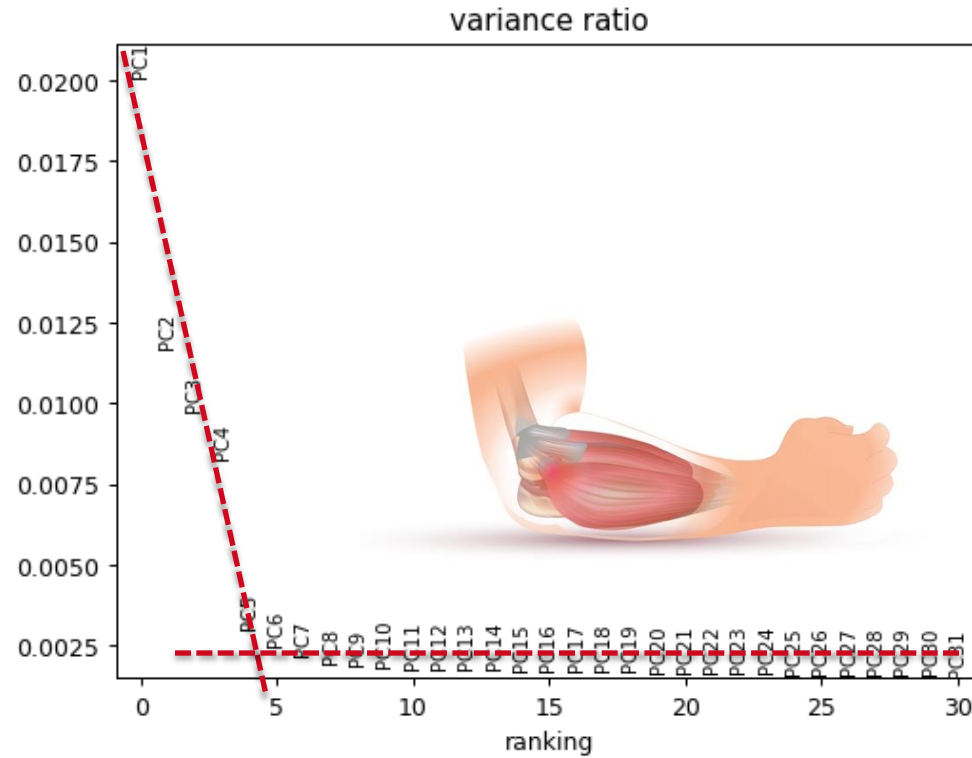
Important loadings on PC2 include NK cell marker NKG7 and B cell marker CD79a

Principal Component Analysis for single cell data



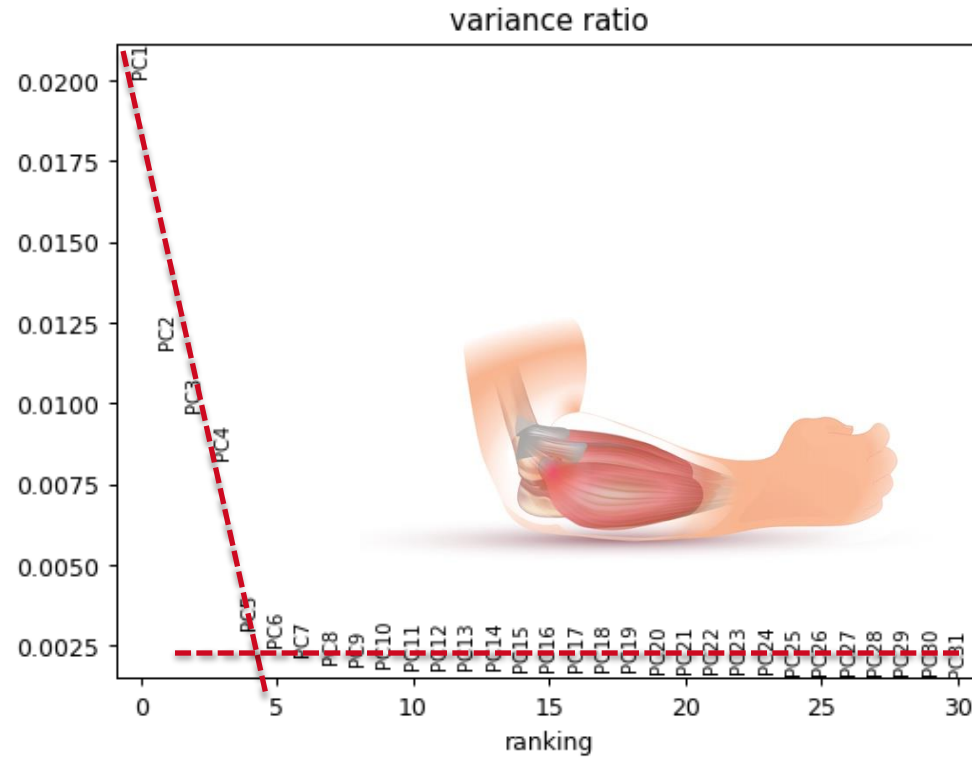


Principal Component Analysis for single cell data





Principal Component Analysis for single cell data

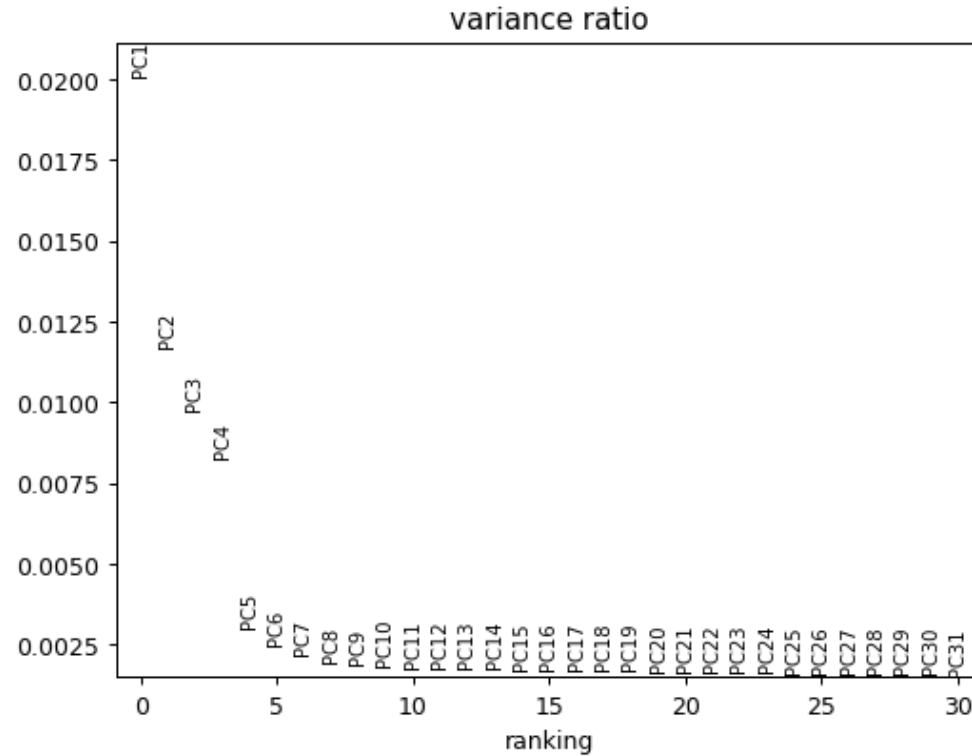


Elbow method can help to decide on the number of PCs to keep ...

- but may be too aggressive
- In this PBMC example, first 100 PCs explain 20% of variance



Principal Component Analysis for single cell data



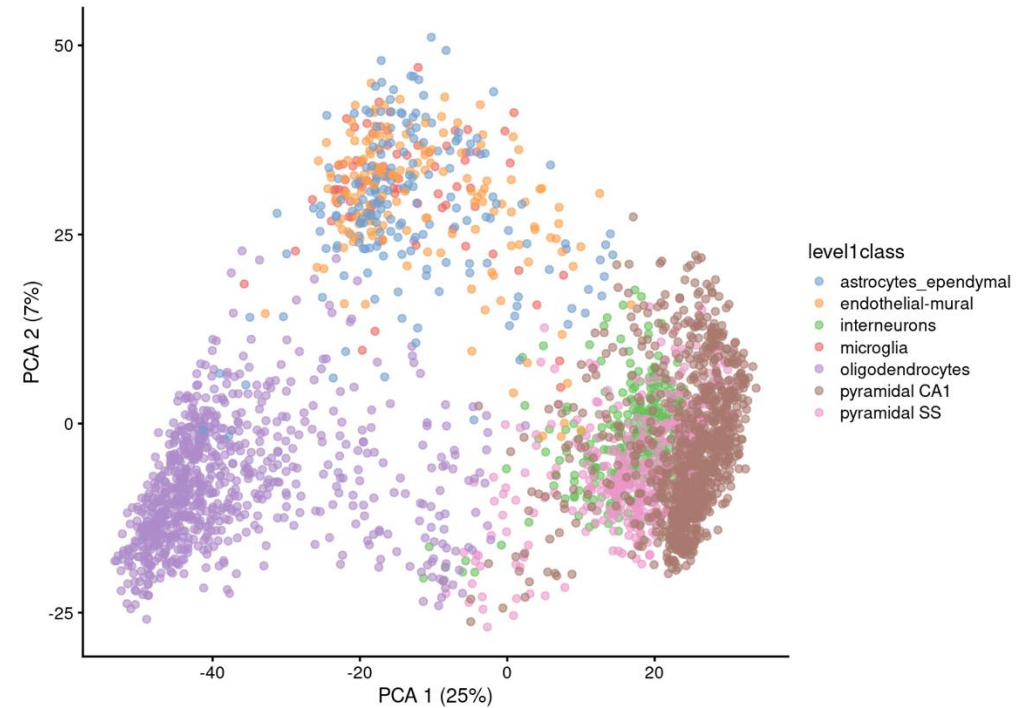
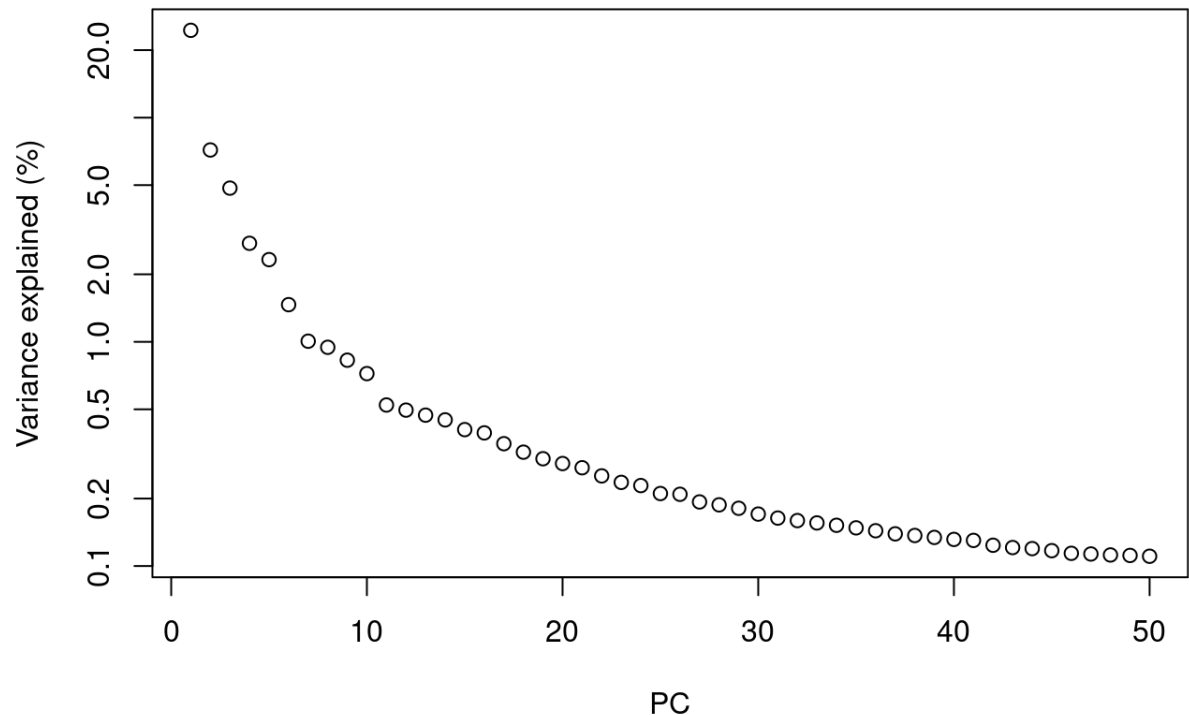
- Typically: keep 10-30 dimensions based on dataset complexity (e.g. number of cell types, conditions)
- If in doubt, keep more rather than removing too much info
- Repeat analysis with different choices of PCs

More diverse populations → first PCs capture more variability



Zeisel 2015 mouse brain dataset

(<https://www.science.org/doi/10.1126/science.aaa1934>, another classic dataset used in many tutorials)



“[...] in the Zeisel dataset, few PCs explain more than 1% of the variance in the entire dataset (Figure 4.1) and choosing between, say, 20 and 40 PCs would not even amount to four percentage points’ worth of difference in variance.”