Dimensionality Reduction method
PCA

PCA
key math (eigenvectors, eigenvalues, variance) is useful.
PCA finds axes (principal components) that maximize variance.
Usage：

1. need a quick, interpretable, linear dimensionality reduction.
2. Common in **gene expression analysis**, where you want to visualize patterns like clusters or remove noise.

Eigenvalues and Eigenvectors

Here is the most important definition in this text.

> ⚠ **Definition.** Let A be an $n \times n$ matrix.
>
> 1. An ***eigenvector*** of A is a *nonzero* vector v in $\mathbf{R}^n$ such that $Av = \lambda v$, for some scalar λ.
> 2. An ***eigenvalue*** of A is a scalar λ such that the equation $Av = \lambda v$ has a *nontrivial* solution.
>
> If $Av = \lambda v$ for $v \neq 0$, we say that λ is the ***eigenvalue for*** v, and that v is an ***eigenvector for*** λ.

Process of PCA

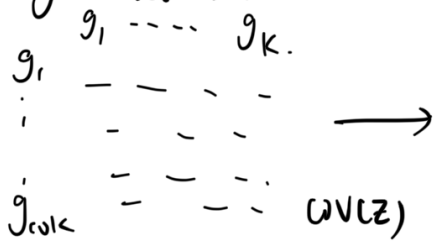
1. standardize the matrix Z (e.g: 100 cell with 10000 genes each)

$Z = \frac{X - \mu}{\sigma}$ (μ, σ are means, standard deviation. column-wise)

2. Compute the covariance Matrix

$COV(Z) = \frac{1}{n-1} Z^T Z$

this generates 10k x 10k matrix.



each entry represents the covariance between 2 genes

3. Eigen Composition → PCs

PCA finds eigenvectors which capture the most variance

eigenvalues. the amount of variance

$COV(Z)\vec{v} = \lambda\vec{v}$, ($\vec{v}$ is one-column vector)

4. select top k PCs

rank the eigenvalues in descending order

choose K eigenvectors to

form a transformation matrix $W_{10000 \times k}$

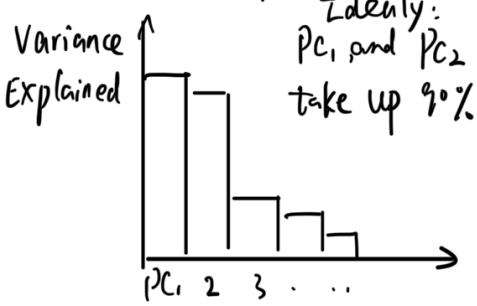
5. project data into PCs. $Z_{PAC} = Z_{100 \times 10000} K_{10000 \times 2}$

(100×2)

PCA Interpretion

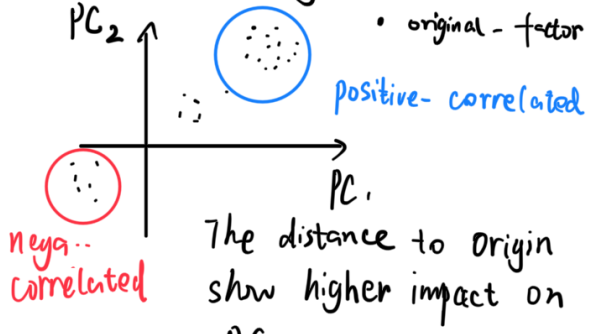
PCA ranking: $PC_1 > PC_2 \cdots PC_n$

screet plot

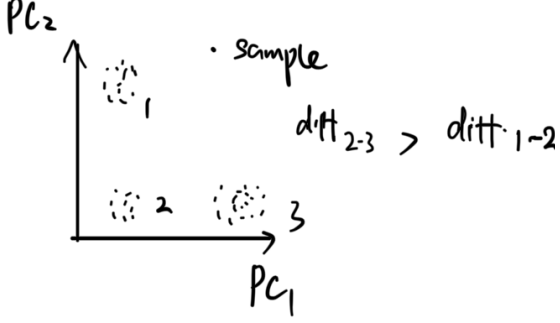


Ideally: PC_1 and PC_2 take up 90%

PCA loadding



The distance to origin show higher impact on PCs.



$diff_{2-3} > diff_{1-2}$

tSNE

This technique is excellent for visualizing complex, high-dimensional data (ex: clusters in gene expression data, single-cell RNA-seq data to identify subpopulations of cells) in two or three dimensions. But, computationally expensive and sensitive to hyperparameters (e.g., perplexity).

t-SNE preserves local structure but not global structure well. Eg: 3 clusters in the final embedding, where 2 are close together and 1 is far away. This **does not** mean they were also far away from each other in the original space. So is less stringent on placing cells far apart that are very different meaning that the distance between clusters is usually meaningless.

It does this by minimising the KL-divergence between the before and after datapoint distributions, P and Q. While t-SNE's goal is not to just minimize dimensions for the sake of reducing size (like PCA), but to preserve the **local structure** of the data—i.e., the **similarity between data points**. It is an **iterative optimization** algorithm, so it starts from an initial **embedding** (an initial low-dimensional representation of the data) and then tries to improve this representation by minimizing a **cost function** that measures how well the low-dimensional representation reflects the pairwise similarities of the original high-dimensional data

Two common initialization strategies in t-SNE:

- **"random" initialization**:
- This means that the starting positions of the points are randomly placed, and t-SNE will try to adjust them during optimization.
- **Pros**: It's a simple, general-purpose initialization.
- **Cons**: Because the initialization is random, it can sometimes result in poor convergence or suboptimal results (not the best representation)
- **"PCA" initialization**:
- It starts with the projection of the data into a lower-dimensional space using **PCA** as the initialization.
- **Pros**: Since it captures the largest variance in the data, it often provides a more sensible starting point for t-SNE, particularly for data where global structure matters
- **Cons**: PCA only captures linear relationships, so it might not be as effective when the data has complex, non-linear relationships.

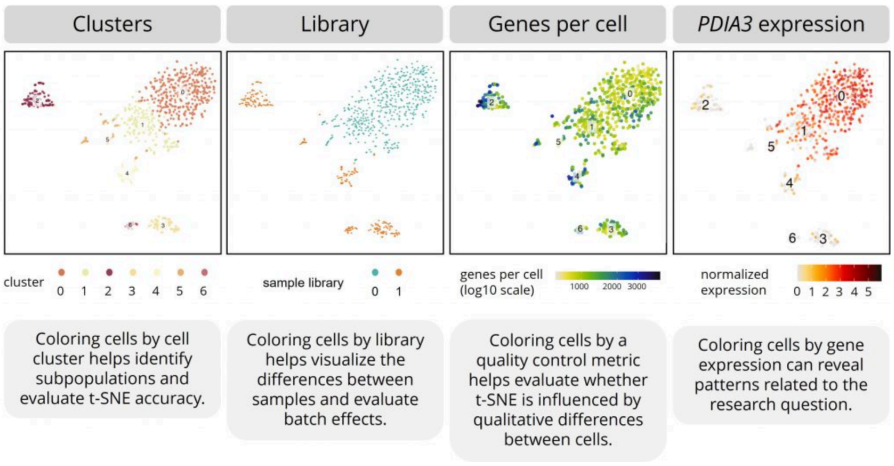


Image: exemplary t-SNE plots with cells colored by cluster, sequencing library, the quality control metric 'genes per cell', and the gene expression pattern of gene PDIA3.

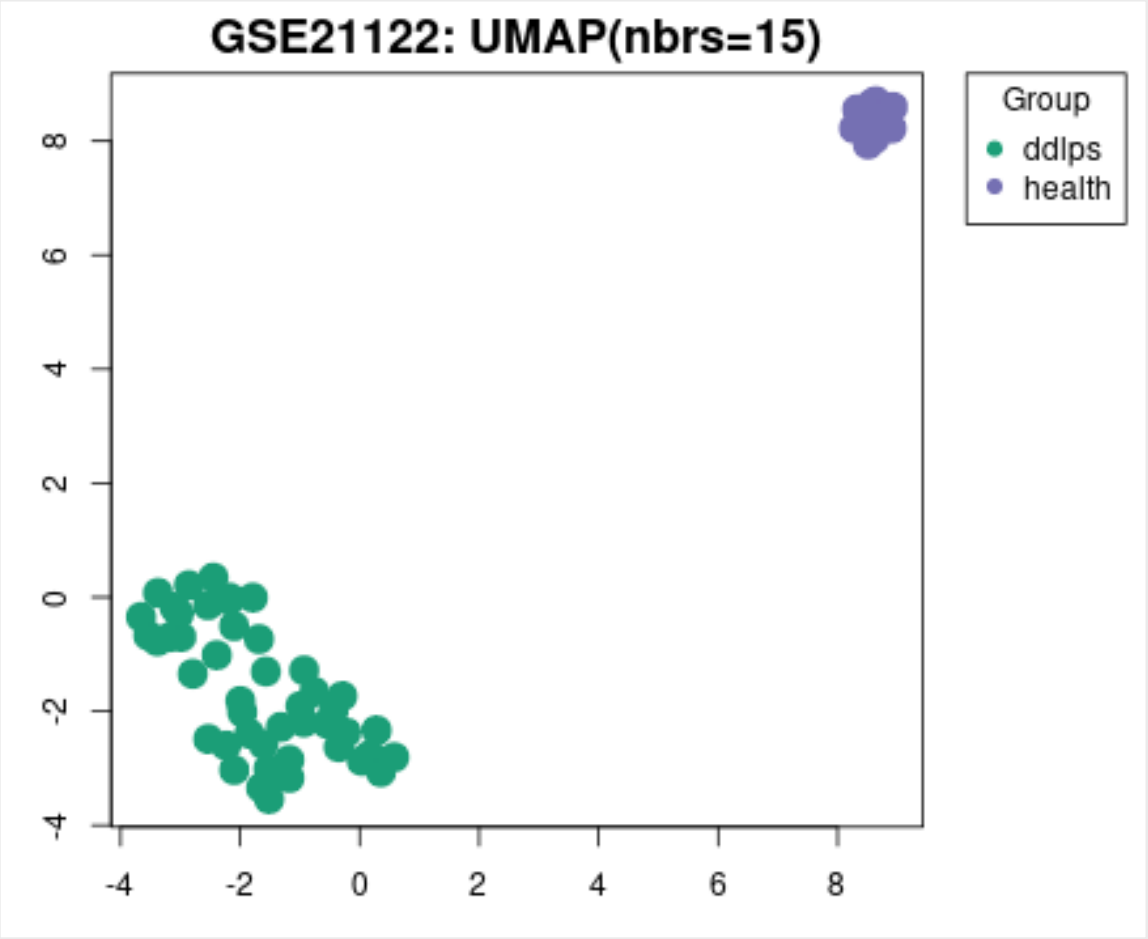
How coloring can represent the purpose of research

UMAP

UMAP:

- Use when you want something similar to t-SNE but with better scalability (ML algorithms' scalability refers to their capacity to handle bigger datasets and computational resources while producing correct results in an acceptable period of time) and preservation of both **local and global structure**.
- Suitable for embedding large datasets like protein sequences or transcriptomics data.

UMAP balances local/global relationships
UMAP can have parameters like n_neighbors.


GSE21122: UMAP(nbrs=15)
Group
ddlps
health
-4
-2
0
2
4
6
8