PCA, t-SNE, UMAP – Differences and Applications

by Jiajia Li

Research School of Biology

14 Aug 2025

Australian National University

Dimensionality Reduction

PCA, t-SNE, and UMAP are all dimensionality reduction techniques, but when do you use it, and which one do you choose?

Before we talk about the technical part of each algorithm, let's talk about dimensionality reduction first.

What is dimensionality reduction?

Dimensionality reduction is all about simplifying data while keeping as much information as possible. Think of it like taking a high-resolution photograph and resizing it – you lose some pixels, but ideally, you keep the important features, so the pictures still makes sense.

Dimensionality: The number of features (genes in our case) describing your data.
Reduction: Shrinking that number of features while preserving structure, relationships, or patterns.

Dimensionality Reduction

Why do it?

- **Efficiency**: Fewer features mean faster computations and less storage.
- **Noise reduction**: Irrelevant or redundant features can drown out meaningful patterns.
- **Visualisation**: It's impossible to visualise data in 100 dimensions, but we can in 2 or 3.

Two main approaches to reduce dimension:

- **Feature selection**: Choose a subset of the original features (we pick top 2000 highly variable genes).
- **Feature extraction**: Create new features that summarise the original ones.
  - **PCA**: Projects data onto directions of maximum variance.
  - **t-SNE**: Great for visualising complex data in 2D or 3D, preserves local neighbourhoods.
  - **UMAP**: Like t-SNE but faster and better at preserving global structure.

PCA

- Projects data onto new axes (principal components) that capture **maximum variance**.
- **Linear method**, best for continuous data with linear relationships.

Typical usage:

- **Preprocessing** before applying machine learning methods (e.g., reducing 1000 features to 50 before clustering or classification). In our case, **cell clustering** is using a machine learning model.
- **Visualisation** in 2D or 3D scatter plots (using first 2 or 3 components). **Not working** when the first 2 or 3 components do not capture the majority of variance.

Examples:

- Reducing dimensionality in gene expression data.
- Speeding up algorithms like SVM or k-means on large feature sets.

PCA

So, the purpose of us running PCA and choose the number of PCs to use is for input into the cell clustering machine learning model (KNN or SNN).

Originally, we have more than 18,000 features (genes). Then, we picked top 2000 highly variable features.

Technically, we can use the top 2000 features as input to the clustering model to find clusters. But that would be too many dimensions, and too much computation and normal computers couldn't handle. It also includes a lot more noise and could cause overfitting.

What if we just pick top 50 highly variable features/genes to find clusters?

Technically, you could, and it is also achievable in code. But 50 genes are just not enough to describe all our cells, right? PCA allows us to reduce the number of features but also keep enough information that describe our data.

Original gene expression matrix

	Cell 1	Cell 2	Cell 3	Cell 4	...	Cell 20000
Gene 1						
Gene 2						
Gene 3						
Gene 4						
Gene 5						
...						
Gene 18000						

Matrix with top 2000 highly variable genes

	Cell 1	Cell 2	Cell 3	Cell 4	…	Cell 20000
Top Gene 1						
Top Gene 2						
Top Gene 3						
Top Gene 4						
Top Gene 5						
…						
Top Gene 2000						

PC Matrix/Embeddings

	Cell 1	Cell 2	Cell 3	Cell 4	…	Cell 20000
PC 1						
PC 2						
PC 3						
PC 4						
PC 5						
…						
PC 50						

t-SNE (t-distributed Stochastic Neighbour Embedding)

- Non-linear technique focusing on **preserving local structure** (similar data points remain close).
- Great for **visualisation** in 2D or 3D, not for general purpose dimensionality reduction for modelling.

Typical usage:

- Visualising high-dimensional data clusters in 2D.
- Exploring patterns and structure in data for Exploratory Data Analysis.

Note:

- Computationally expensive.
- Non-deterministic, results can vary run to run unless you fix random seed.
- Distances in the plot are not globally meaningful, good for neighbours, not large-scale distances.

UMAP (Uniform Manifold Approximation and Projection)

- **Non-linear method** like t-SNE but faster and better at preserving both **local and global structure**.
- Scales better for large datasets.

Typical usage:

- **Visualisation** (like t-SNE) but on bigger datasets.
- Preprocessing for clustering or even feeding into models, better than t-SNE for this because it preserves more structure.

Advantages over t-SNE:

- Faster and more scalable.
- Captures both local neighbourhoods and overall data shape better.
- Can be used for general dimensionality reduction, not just visualisation.

UMAP (Uniform Manifold Approximation and Projection)

So, back to the question we had in class:

Does the `RunUMAP()` function calculates on the gene expression matrix table or the PC embeddings?

The answer is on the PC embeddings!

Because we have used the PC embeddings to find clusters in our cell, when we need to project the cells to 2D to visualise it, we need to use the same information.

Technically, you can run UMAP on the gene matrix table or any types of matrix.

Thank you

Contact us

Jiajia Li
Research School of Biology

RN Robertson Building, 46 Sullivan's Creek Rd
The Australian National University
Canberra ACT 2600

E jiajia.li1@anu.edu.au

Australian National University

TEQSA PROVIDER ID: PRV12002 (AUSTRALIAN UNIVERSITY)
CRICOS PROVIDER CODE: 00120C