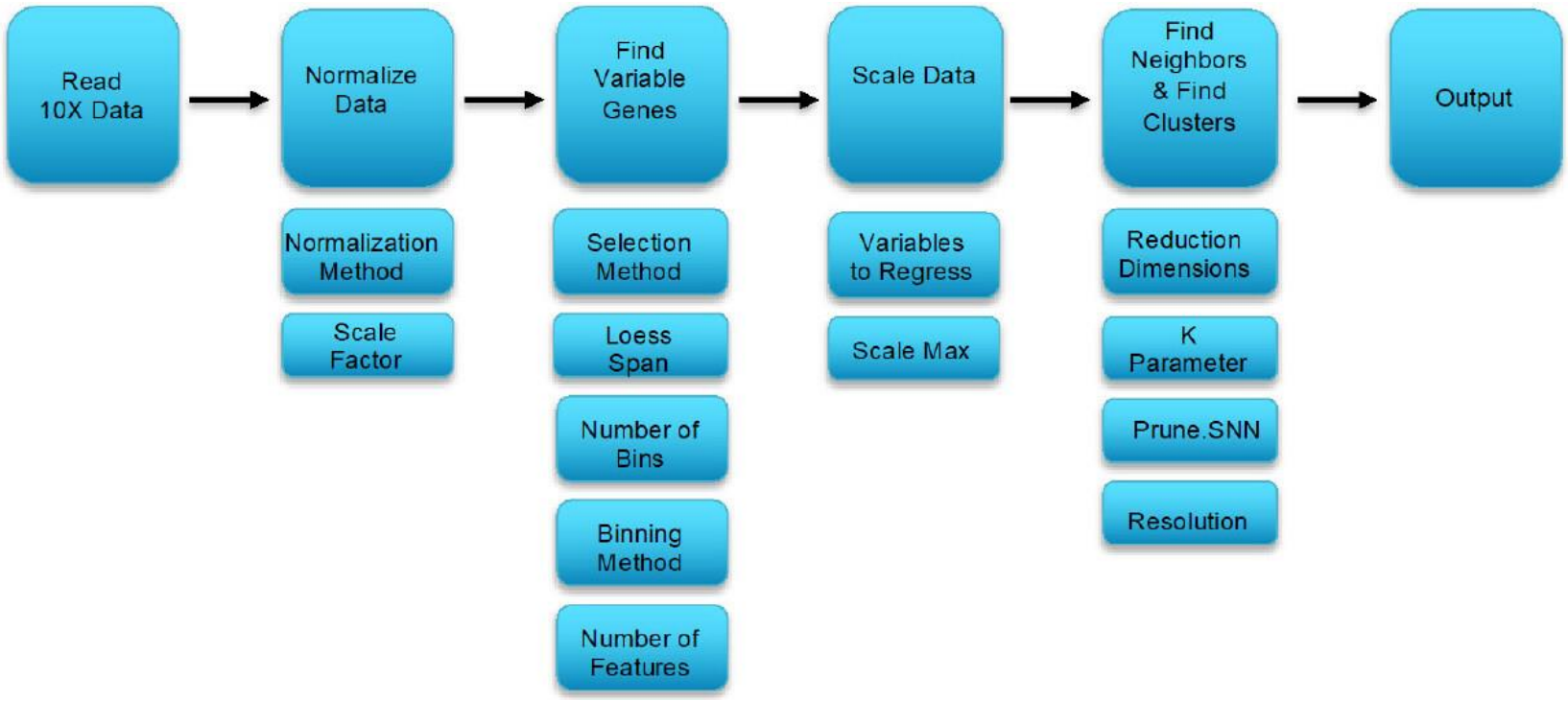
Lecture 4
scRNA-seq analysis workflow, normalization, visualization

Outline

- Standard scRNA-seq data analysis workflow: Seurat and Scanpy
- scRNA-seq normalization, highly variable gene selection
- Dimensional reduction, visualization

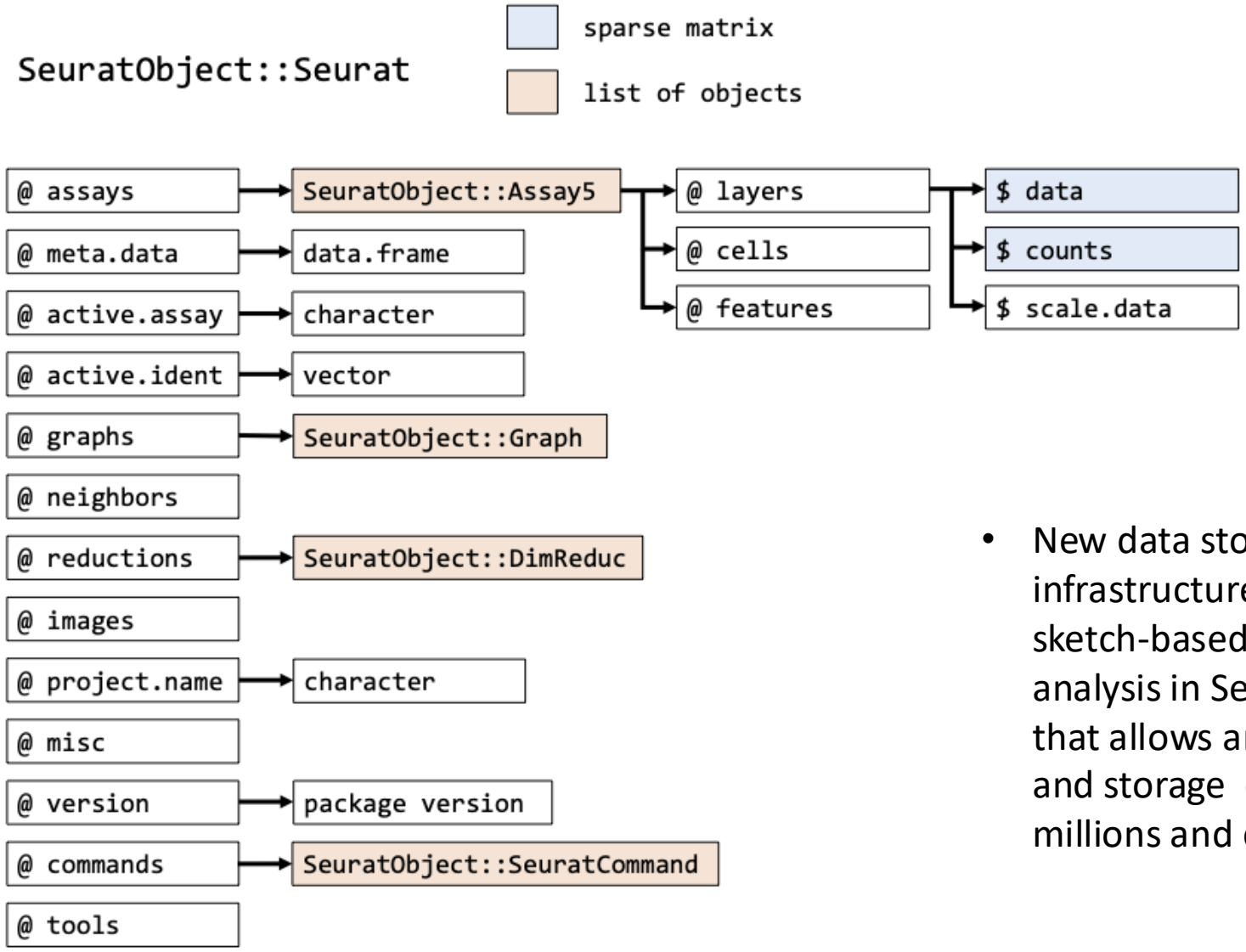
Seurat (Satija group)

- An R package that is widely used
- Current version v5 supports multi-modality and scalable analysis



Schneider, I., Cepela, J., Shetty, M., Wang, J., Nelson, A. C., Winterhoff, B., & Starr, T. K. (2021). Use of “default” parameter settings when analyzing single cell RNA sequencing data using Seurat: a biologist’s perspective. *J Transl Genet Genom*, *5*, 37-49.

Seurat object

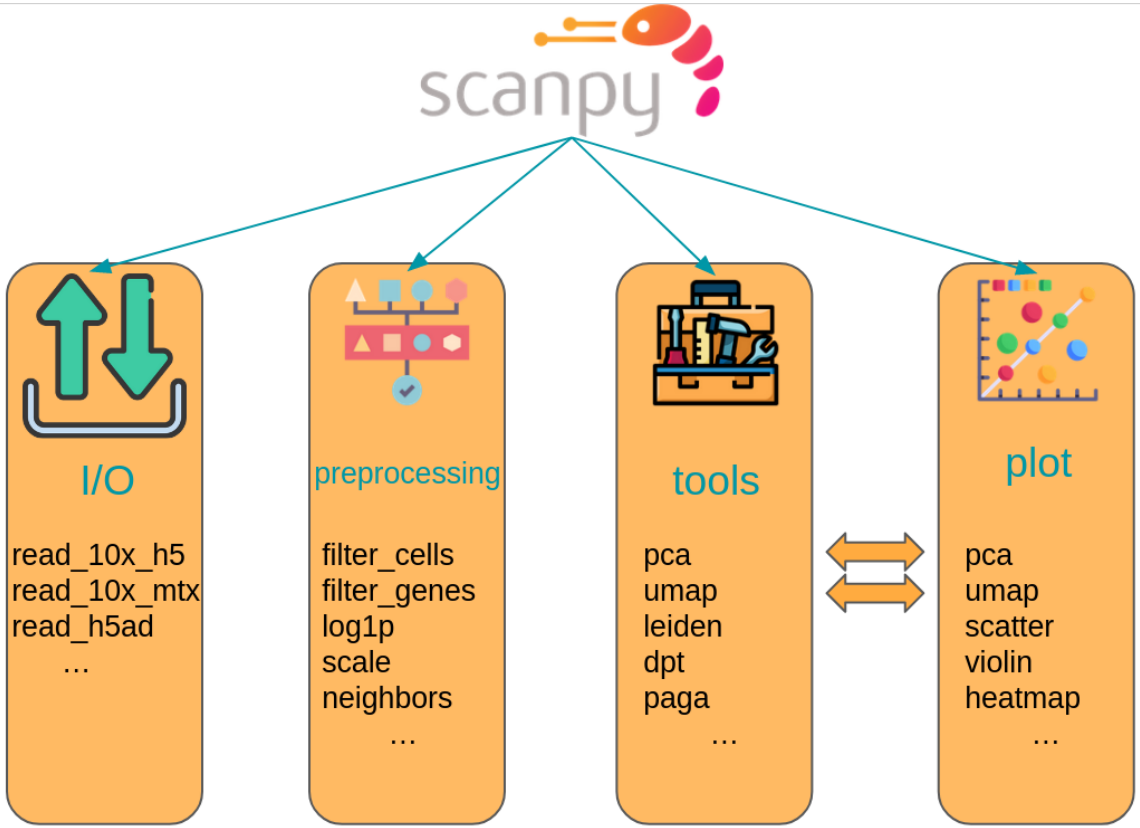


Reference tutorial: https://sib-swiss.github.io/single-cell-training/day1/day1-2_analysis_tools_qc.html

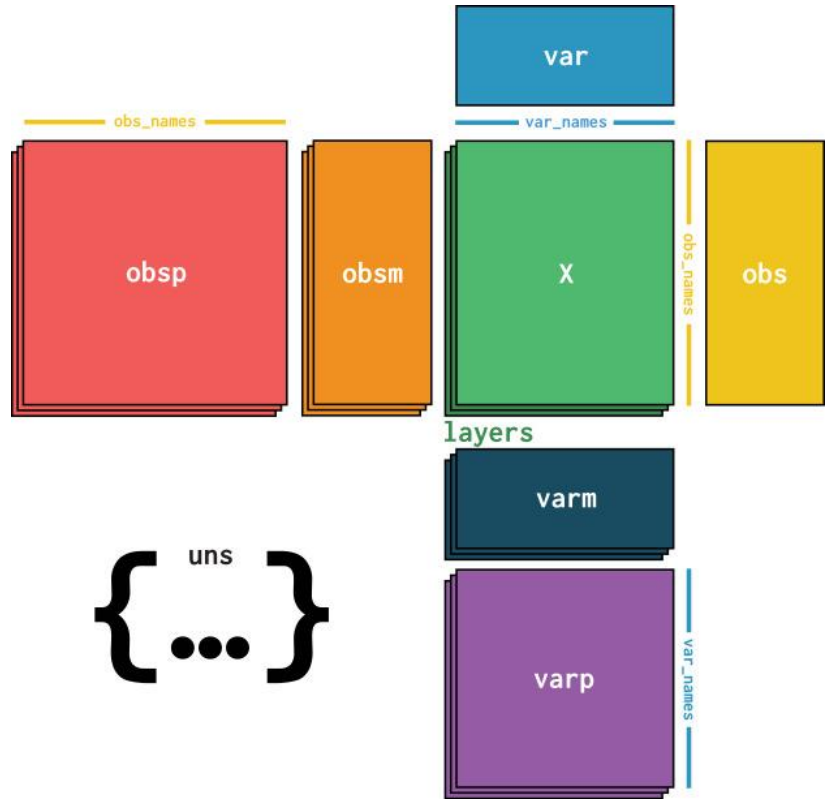
- New data storage infrastructure and sketch-based analysis in Seurat v5 that allows analysis and storage of millions and cells

Scanpy (Wolf et. al. Genome Biology 2018)

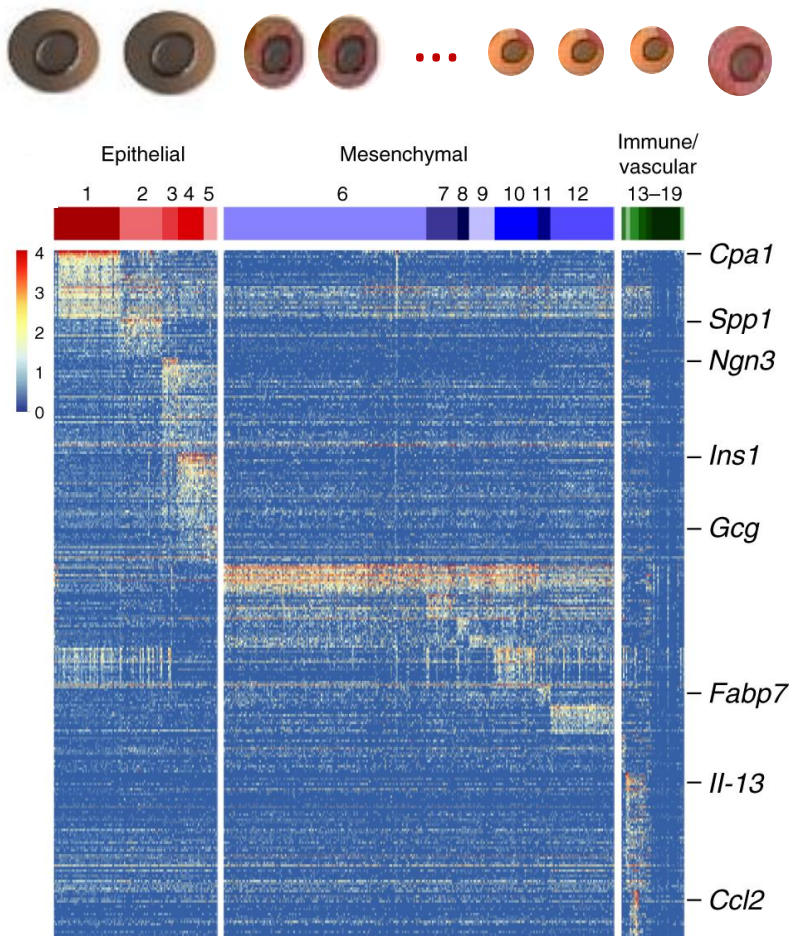
- A python package alternative to Seurat
- Handle large-scale data
- Easy to interface with deep-learning based methods



AnnData object



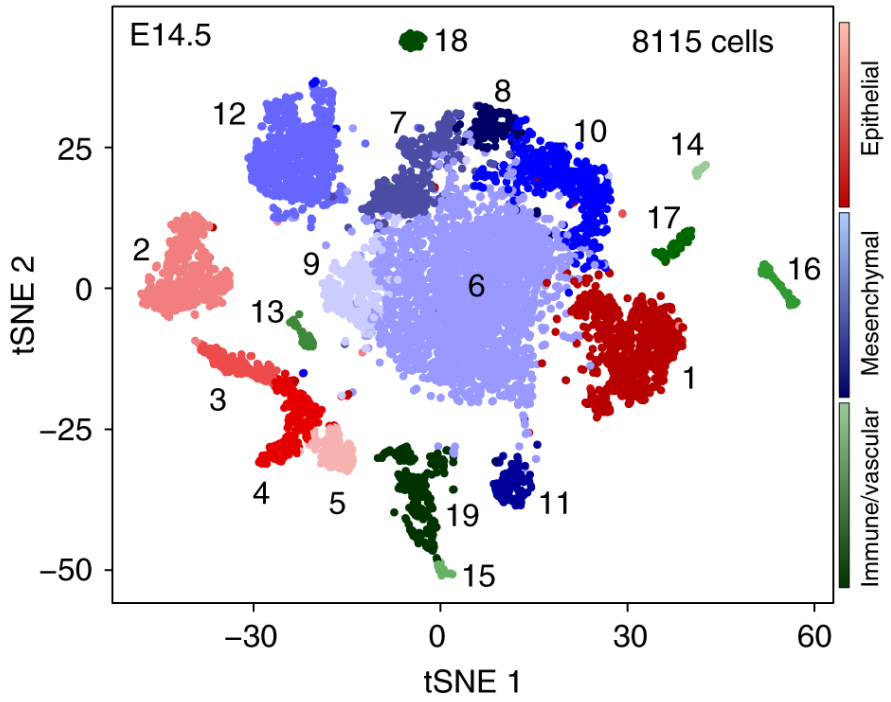
scRNA-seq dimension reduction and visualization



Genes

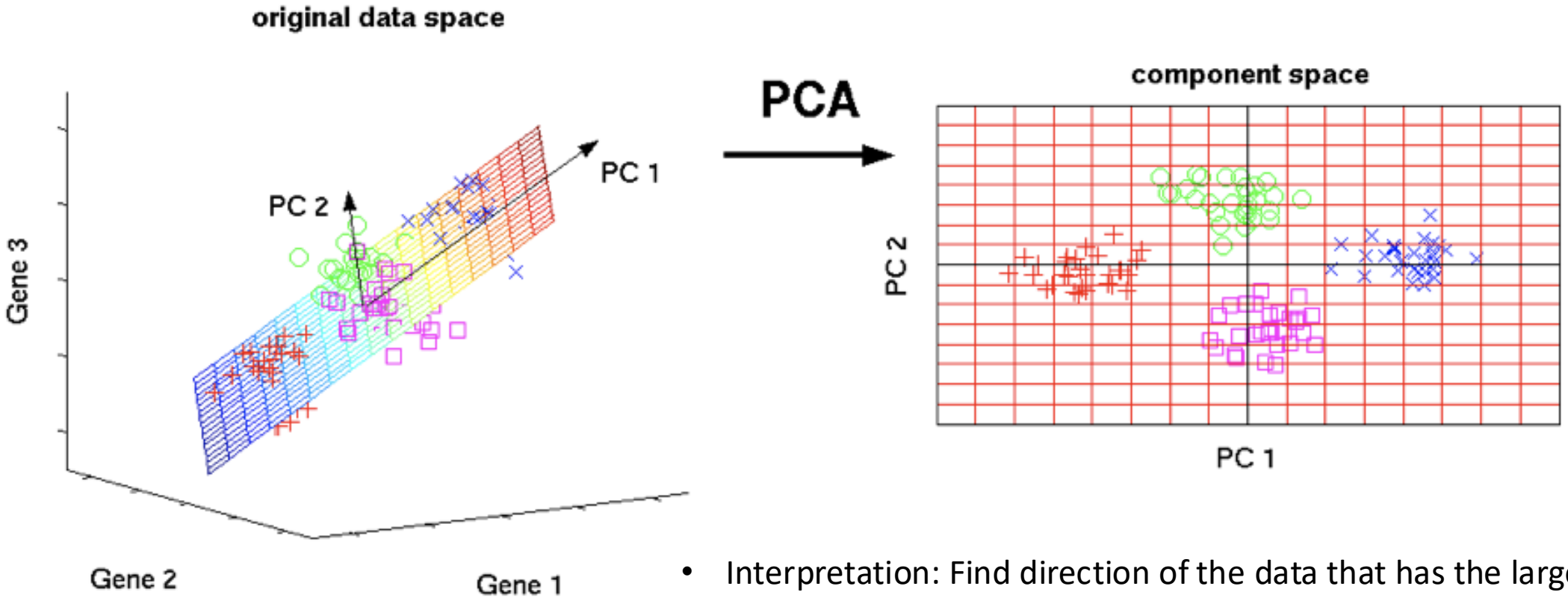
Cells

(Colors are determined by separate clustering methods!!)



Lineage dynamics of murine pancreatic development at single-cell resolution, Byrnes et. al. *Nature Comm.* 2018

Linear dimension reduction: PCA



- Interpretation: Find direction of the data that has the largest variation
- Not ideal for visualization

- Requires proper normalization of the data for using Euclidean distance
- High-dimensional PCA is not accurate

scRNA-seq normalization

Why do we need normalization?

- Raw counts across cells are not comparable → adjust for library size
- Make the data more "Gaussian" before using linear methods like PCA

Shifted logarithm

- Library size normalization + taking logarithm

$$f(Y_{gc}) = \log(\frac{Y_{gc}}{s_c} + y_0)$$

- y_0: pseudo-count to avoid $\log(0)$. Typically $y_0 = 1$ to make the normalized data sparse
- $s_c = l_c/L$ so that y_0 is not too influential. $L = 10^4$ (Seurat and Scanpy default)

- Shifted logarithm is approximately doing some variance stabilization

$$\mathrm{var}\left(f(Y_{gc})\right) \approx \frac{s_c^2}{\mu_g^2}\mathrm{var}(Y_{gc})$$

if $Y_{gc} \sim NB(\mu_g, \theta)$ then $\mathrm{var}(Y_{gc}) = \mu_g + \theta\mu_g^2$, variance stabilized if θ or μ_g is large

- Scaling: standardize each gene across cells to have mean 0 and variance 1 after log-normalization

Pearson / deviance residuals

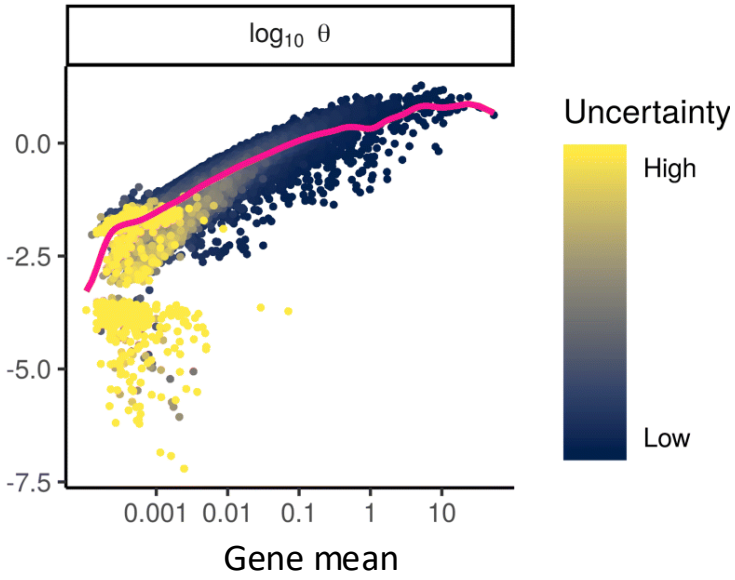
Sctransform (Hafemeister and Satija, Genome Biology 2019; Choudhary and Satija, Genome Biology 2022)

$$
\begin{aligned}
x_{gc} &\sim \mathrm{NB}(\mu_{gc}, \theta_g) \\
\ln \mu_{gc} &= \beta_{g0} + \ln n_c,
\end{aligned}
$$

- n_c is the library size, x_{gc} is the observed count (Y_{gc})
- Assume $\mu_{gc} = n_c p_g$

- Pearson residual normalization

$$
\begin{aligned}
Z_{gc} &= \frac{x_{gc} - \mu_{gc}}{\sigma_{gc}} \\
\mu_{gc} &= \exp \beta_{g0} + \ln n_c \\
\sigma_{gc} &= \sqrt{\mu_{gc} + \frac{\mu_{gc}^2}{\theta_{gc}}},
\end{aligned}
$$



- Estimate θ_g as a smoothed function of μ_g. $\theta_g = \infty$ for small μ_g

- If we are interested in heterogeneity across cells, then μ_{gc} contains non-interesting information
- Normalized data is not sparse any more

Pearson / deviance residuals

Deviance residuals (Townes et. al. Genome Biology 2019)

- For general definitions, check a GLM book
- The deviance residuals can look more normal than Pearson residuals

- Assume Poisson model on the observed counts

$$Z_{cg} = \text{sign}\left(X_{cg} - \hat{\mu}_{cg}\right)\sqrt{2\left[X_{cg}\ln\frac{X_{cg}}{\hat{\mu}_{cg}} - \left(X_{cg} - \hat{\mu}_{cg}\right)\right]}$$

- Assume NB model on the observed counts

$$Z_{cg} = \text{sign}\left(X_{cg} - \hat{\mu}_{cg}\right)\sqrt{2\left[X_{cg}\ln\frac{X_{cg}}{\hat{\mu}_{cg}} - \left(X_{cg} + \theta\right)\ln\frac{X_{cg} + \theta}{\hat{\mu}_{cg} + \theta}\right]}$$

(formula and notations copied from Lause et. al. Genome Biology 2021)

- Assume multinomial distribution (Townes et. al. Genome Biology 2019)

$$r_{ij}^{(d)} = \text{sign}(y_{ij} - \hat{\mu}_{ij})\sqrt{2y_{ij}\log\frac{y_{ij}}{\hat{\mu}_{ij}} + 2(n_i - y_{ij})\log\frac{n_i - y_{ij}}{n_i - \hat{\mu}_{ij}}}$$

  - Almost identical to the Poisson deviance

Selection of highly variable genes (HVG)

- High-dimensional PCA is not accurate when latent factors are not strong enough
- If a gene is expressed homogeneously across cells, it does not contain information about cell heterogeneity and only contribute noise to PCA

- Selection of HVG:
  only use genes that have higher variability across cells than background when doing PCA
  - Identify a subset of 500-2000 genes

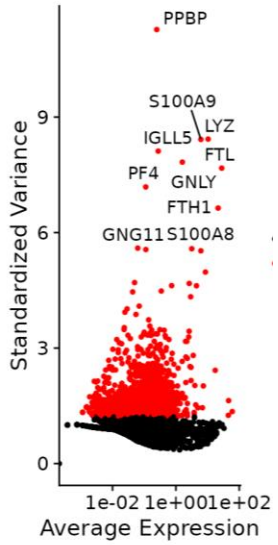
  - Using Sctransform Pearson residuals
    - Calculate variance of Z_{gc} for each g across c, select the top ones

  - Default method in Seurat: same idea, but a more straight-forward way to get Z_{gc}

$$Z_{gc} = \frac{Y_{gc} - \bar{Y}_g}{\sigma_g}$$

  σ_g is calculated by fitting a smoothed mean-variance relationship

  - Calculate residual deviance:
    if Z_{gc} are deviance residuals, rank genes based on $\sum_c Z_{gc}^2$



Non-linear visualization: t-SNE & UMAP

- PCA for dimension reduction:
  - Only use HVG to perform PCA and get PC loadings
  - Selection top k ($k = 50$ in Seurat default) PCs to reduce data dimensions for further cell-level analyses
  - Systematic selection of k is possible but can be time consuming and may not worth it

- t-SNE: t-Distributed Stochastic Neighbor Embedding

  Paper: https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf
  Presentation: https://www.youtube.com/watch?v=RJVL80Gg3lA&list=UUtXKDgv1AVoG88PLl8nGXmw

- UMAP: Uniform Manifold Approximation and Projection

  Paper: https://arxiv.org/pdf/1802.03426.pdf
  Benchmark paper on scRNA-seq: https://www.nature.com/articles/nbt.4314
  Presentation: https://www.youtube.com/watch?v=nq6iPZVUxZU

The idea of t-SNE

SNE (stochastic neighbor embedding)

- Preserve the similarity of high-dimensional points in low-dimensional points
- Measure similarity (conditional distributions) by Gaussian density

Original space:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

Low-dimensional space:

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$

Find $\{y_i\}$ to minimize:

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

- Because of asymmetry in the KL divergence
  - large cost for using widely separated (y_i, y_j) to represent nearby (x_i, x_j)
  - Small cost for using nearby (y_i, y_j) to represent widely seperated (x_i, x_j)
  - Only retain local structure of the data

The idea of t-SNE

SNE (stochastic neighbor embedding)

- Determination of the standard deviations σ_i
  - Smaller σ_i for denser regions and larger σ_i for sparser regions
  - For each I, find σ_i that reaches a pre-specified perplexity

$$Perp(P_i) = 2^{H(P_i)},$$

where $H(P_i)$ is the Shannon entropy of P_i measured in bits

$$H(P_i) = -\sum_i p_{j|i} \log_2 p_{j|i}.$$

  - Decrease perplexity to preserve more global structures

- Solution obtained by gradient descent
  - Initialization: randomly sampled points from independent Gaussian
  - Large momentum to avoid poor local minima
  - Difficult to optimize and has “crowding problem”

The idea of t-SNE

t-SNE (t-distribution density [Cauchy])

Original space:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)} \qquad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

Low-dimensional space:

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$

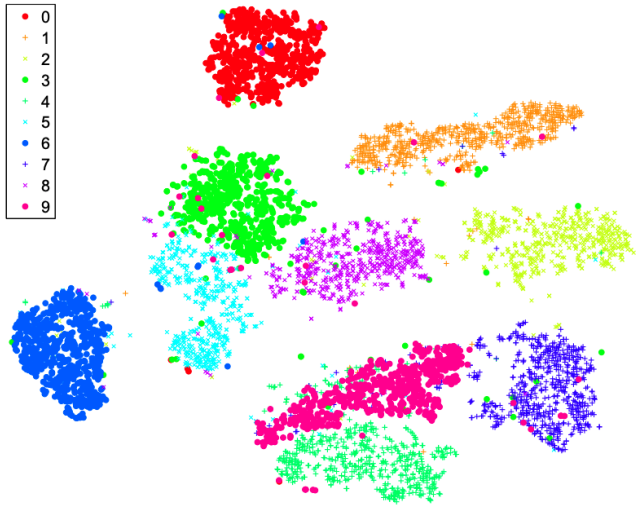
Find $\{y_i\}$ to minimize:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

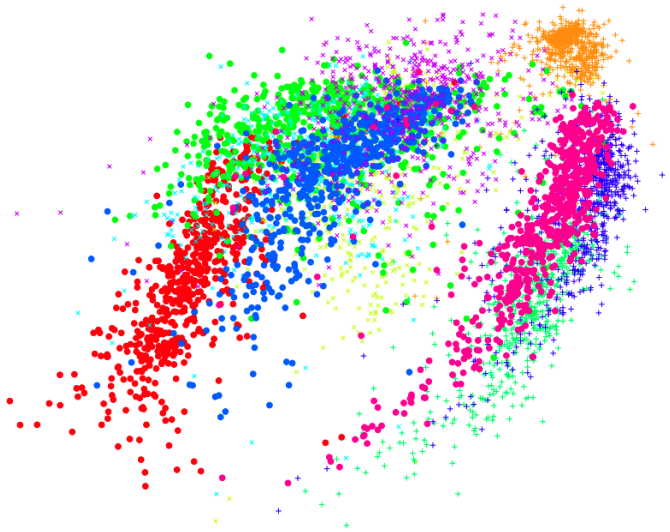
$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

- Represent high-dimensional points better and keep moderately far-away points not too close
- Faster to optimize because calculation does not involve exponential
- Computational cost: $O(n^2)$

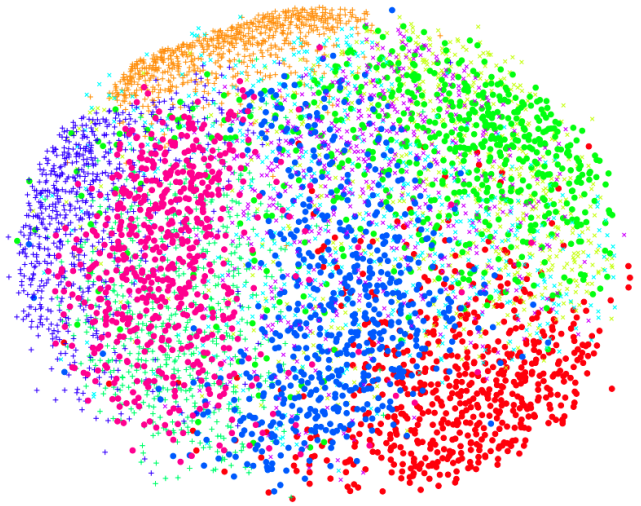
Visualization of MNEST data



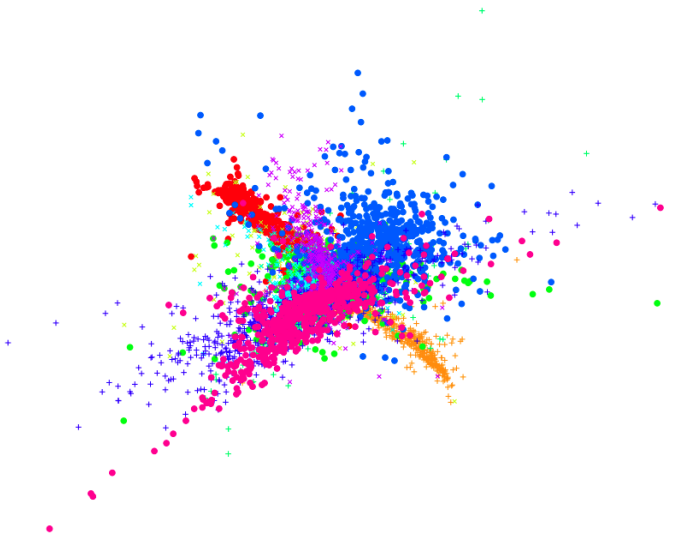
(a) Visualization by t-SNE.



(a) Visualization by Isomap.



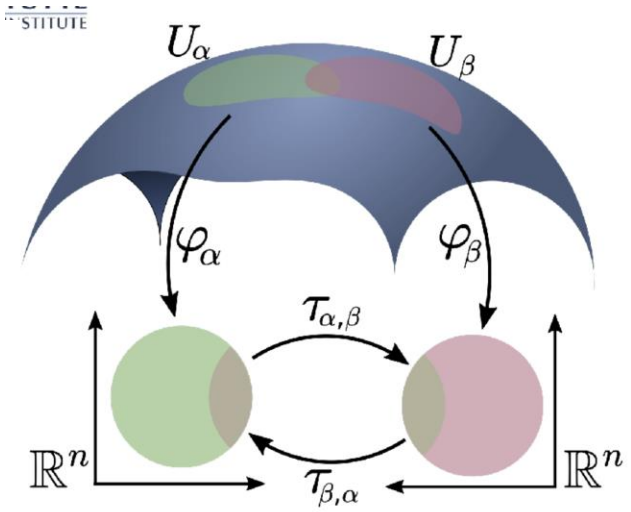
(b) Visualization by Sammon mapping.



(b) Visualization by LLE.

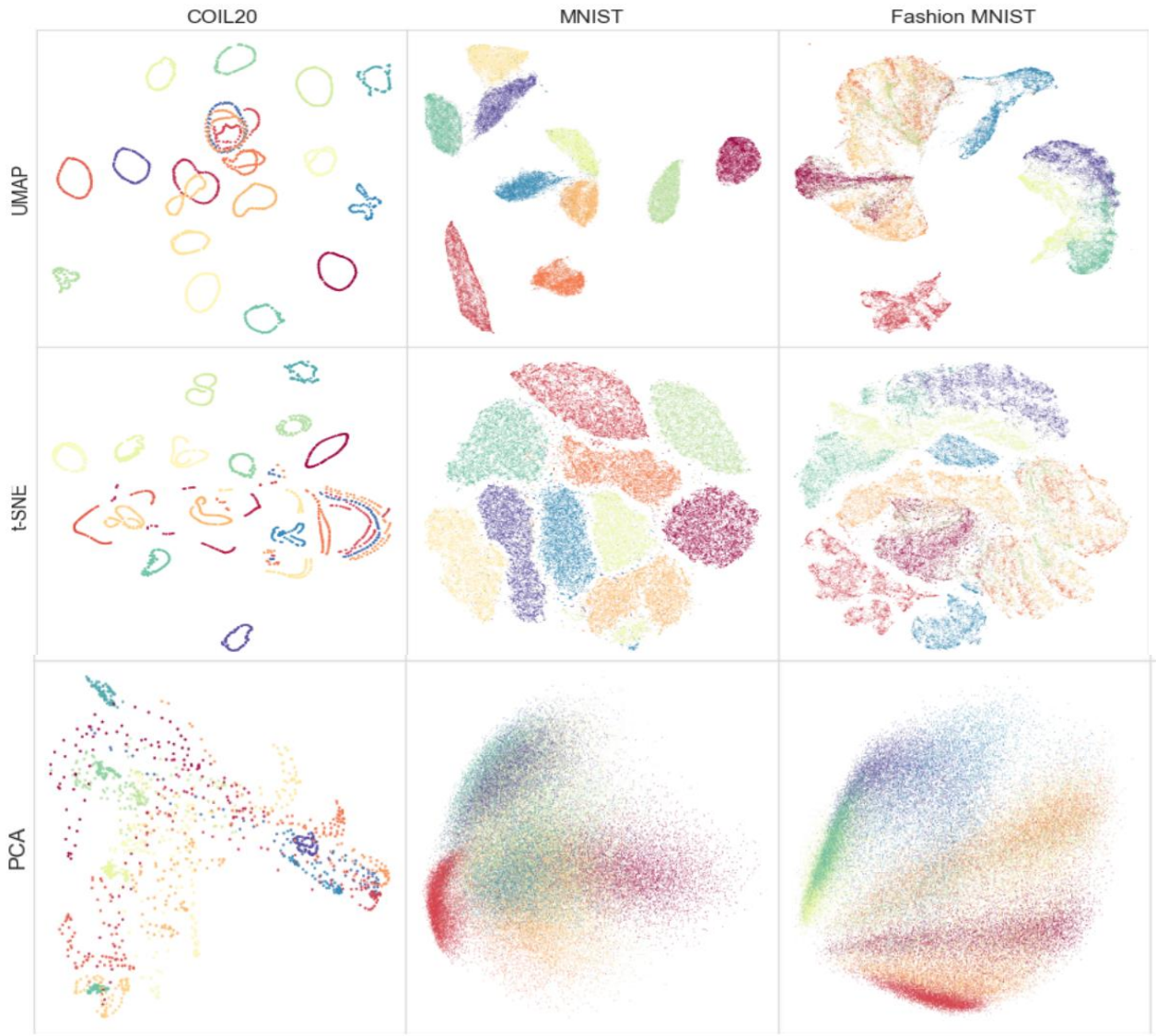
The (very high-level) idea of UMAP

- Construct topological representation of high-dimensional data
  - Assume that the data points uniformly lie on a low-dimensional manifold
  - Define local distance by k-nearest neighbors and construct a weighted k-neighbour graph
  - Based on the theory of local fuzzy simplicial set representations
- Represent the manifold by low-dimensional points
  - Minimize cross entropy of fuzzy simplicial set representation between the low and high-dimensional space
  - Use force-directed graph layout algorithm in low-dimensional space
- Computational cost: $O(n^{1.14})$



https://www.youtube.com/watch?v=nq6iPZVUxZU

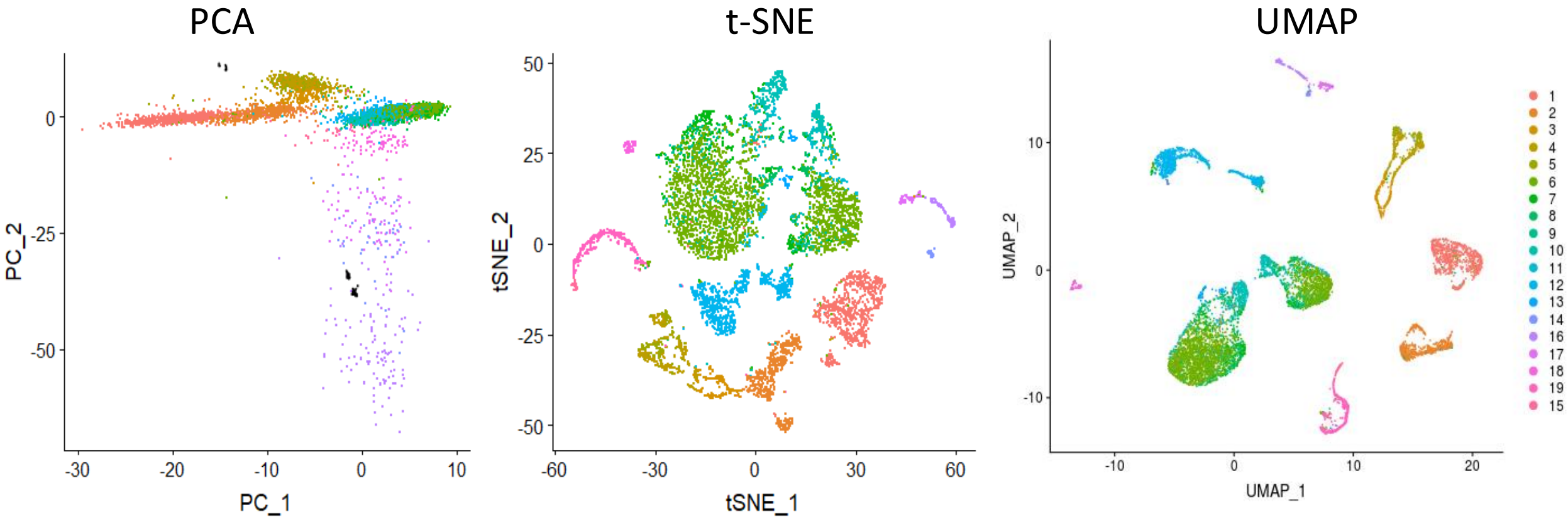
Compare PCA, t-SNE, UMAP



- PCA: keep global distance
- T-SNE: focus on local distance
- UMAP: focus on local distance, but may keep more global distance features

https://arxiv.org/pdf/1802.03426.pdf

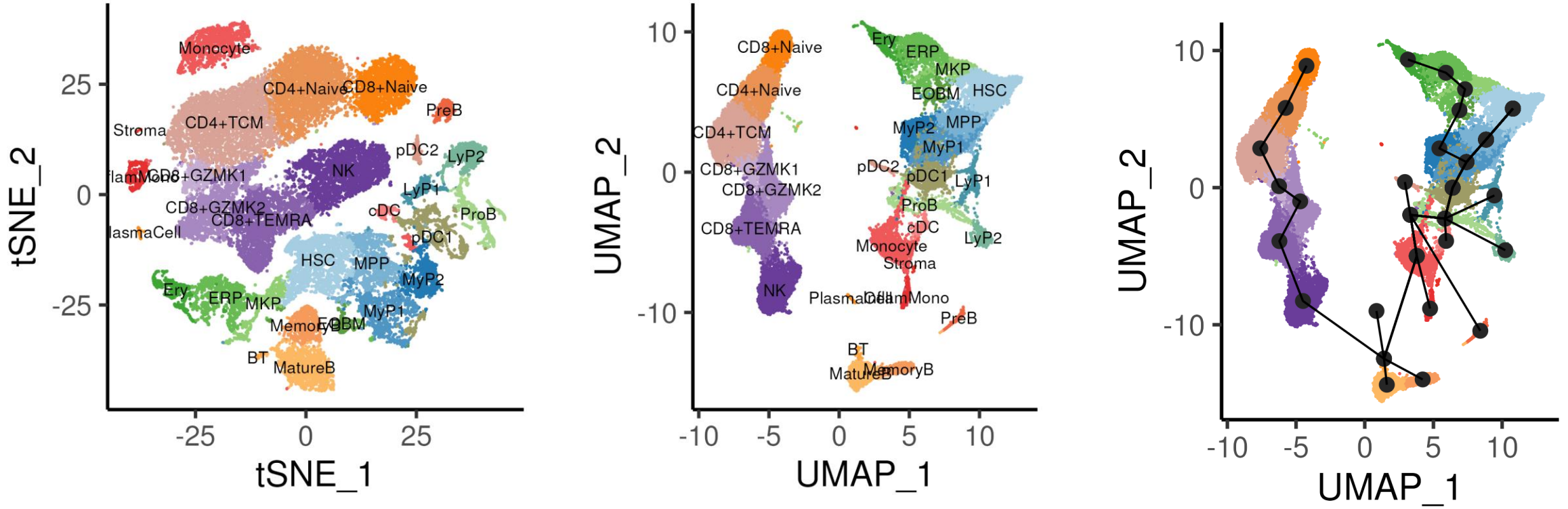
Visualize scRNA-seq using PCA, t-SNE, UMAP



Data from paper: Lineage dynamics of murine pancreatic development at single-cell resolution, Byrnes et. al. *Nature Comm.* 2018

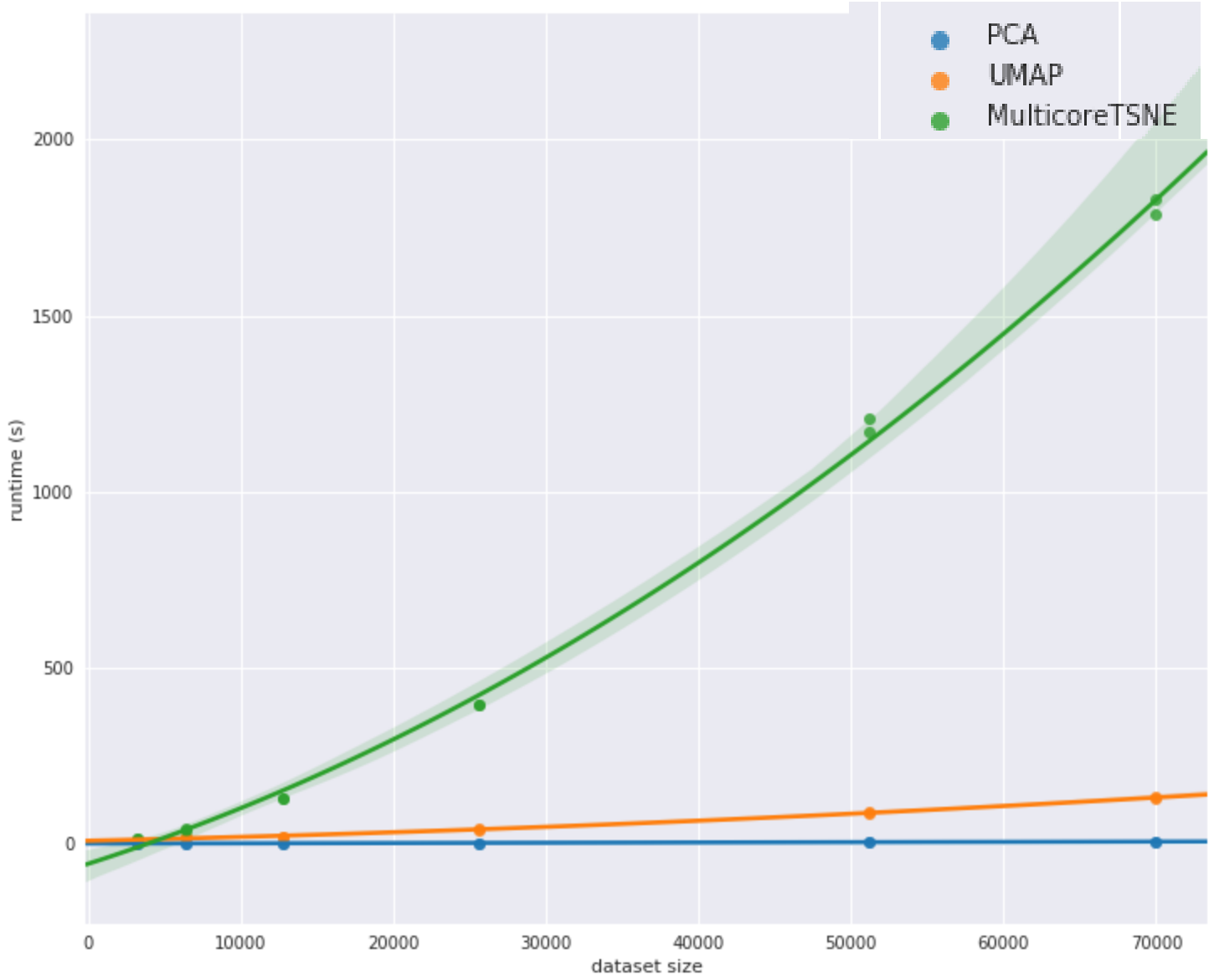
Analysis pipeline see Seurat tutorial: https://satijalab.org/seurat/v3.0/pbmc3k_tutorial.html

UMAP is better at showing the cell lineages



https://ouyanglab.com/singlecell/dimrd.html#trajectory-inference-and-pseudotime

Running time comparison



- Computation of UMAP is based on the construction of k-nearest-neighbor graph
- Nearest neighbors are obtained using the top PCs
- Computational cost: $O(n^{1.14})$

https://umap-learn.readthedocs.io/en/latest/benchmarking.html

Related papers

- Wolf, F. A., Angerer, P., & Theis, F. J. (2018). SCANPY: large-scale single-cell gene expression data analysis. Genome biology, 19, 1-5.
- Hafemeister, C., & Satija, R. (2019). Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. Genome biology, 20(1), 296.
- Choudhary, S., & Satija, R. (2022). Comparison and evaluation of statistical error models for scRNA-seq. Genome biology, 23(1), 27.
- Townes, F. W., Hicks, S. C., Aryee, M. J., & Irizarry, R. A. (2019). Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. Genome biology, 20, 1-16.
- Lause, J., Berens, P., & Kobak, D. (2021). Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. Genome biology, 22, 1-20.