STAT 35510
Lecture 5

Spring, 2024
Jingshu Wang

Outline

- scRNA-seq denoising methods

- Trajectory analysis

scRNA-seq denoising

- scRNA-seq data is very noisy

$$\begin{tikzcd}\text{Observed count data } Y_{gc} & \text{True gene expression } X_{gc} \arrow[l]\end{tikzcd}$$

$$\begin{array}{c|ccccc}
 & \text{Cells} \longrightarrow & & & & \\
\hline
\text{Genes} \downarrow & 0 & 3 & 0 & \cdots & 0 \\
 & 4 & 0 & 1 & \cdots & 0 \\
 & 12 & 10 & 4 & \cdots & 1 \\
 & 0 & 0 & 0 & \cdots & 0
\end{array}$$

$\longleftrightarrow$

biological variation + measurement error

$$\begin{array}{c|ccccc}
 & \text{Cells} \longrightarrow & & & & \\
\hline
\text{Genes} \downarrow & 0 & 12 & 0 & \cdots & 30 \\
 & 15 & 7 & 1 & \cdots & 0 \\
 & 120 & 147 & 62 & \cdots & 27 \\
 & 0 & 0 & 0 & \cdots & 5
\end{array}$$

$\longleftrightarrow$

biological variation

Data denoising: get an estimate of X

Core idea:

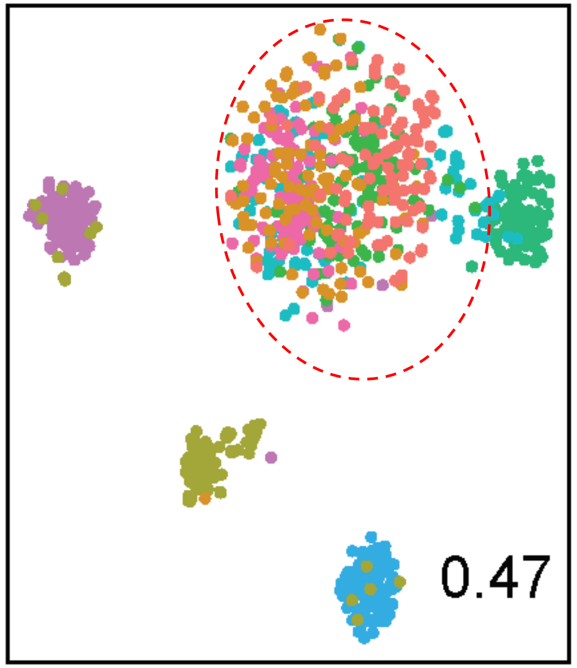
- Use gene-gene dependence or cell-cell similarity to remove noise
  - “smooth” over similar genes or similar cells

- Denoising is also described as “imputation”, however this is NOT a missing data problem!

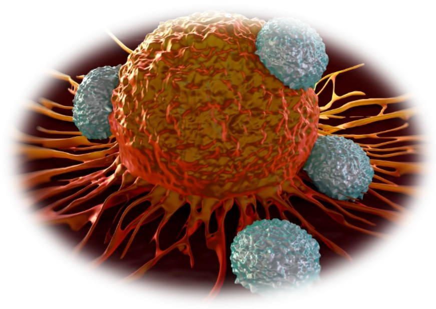
How can denoising help?

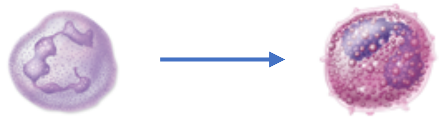
900 PBMC cells (immune cells in peripheral blood) with labels [Zheng et. al., 2017]

Original

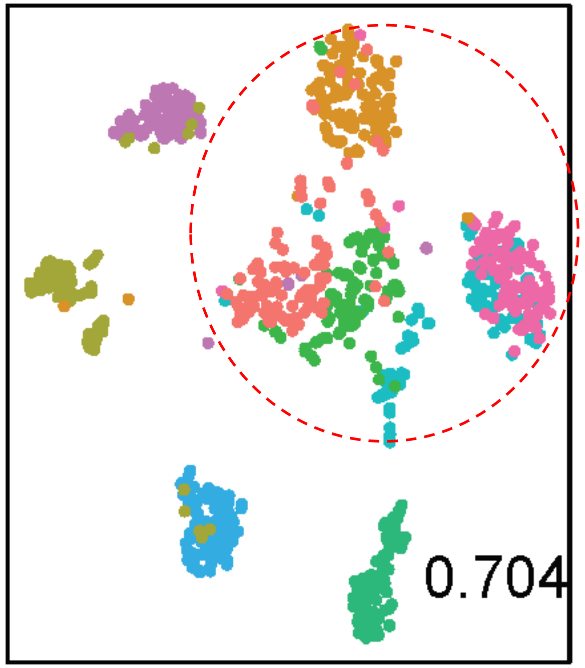


T cell





After Denoising



B cells ● Monocytes ● CD34+ ● NK cells ●

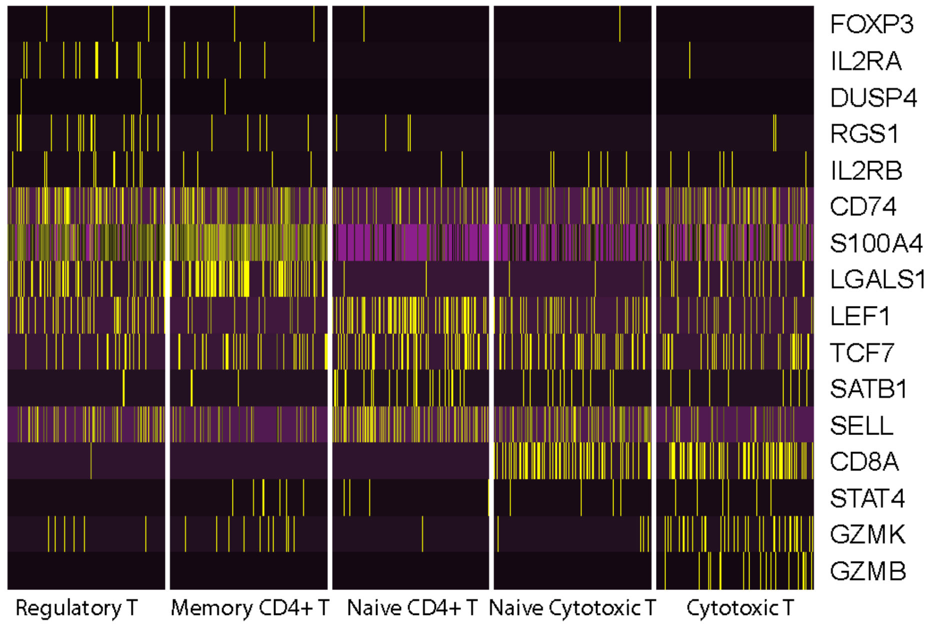
T cells

CD4+ Memory ● Regulatory ● CD4+ Naive T ● Naive cytotoxic ● cytotoxic ●

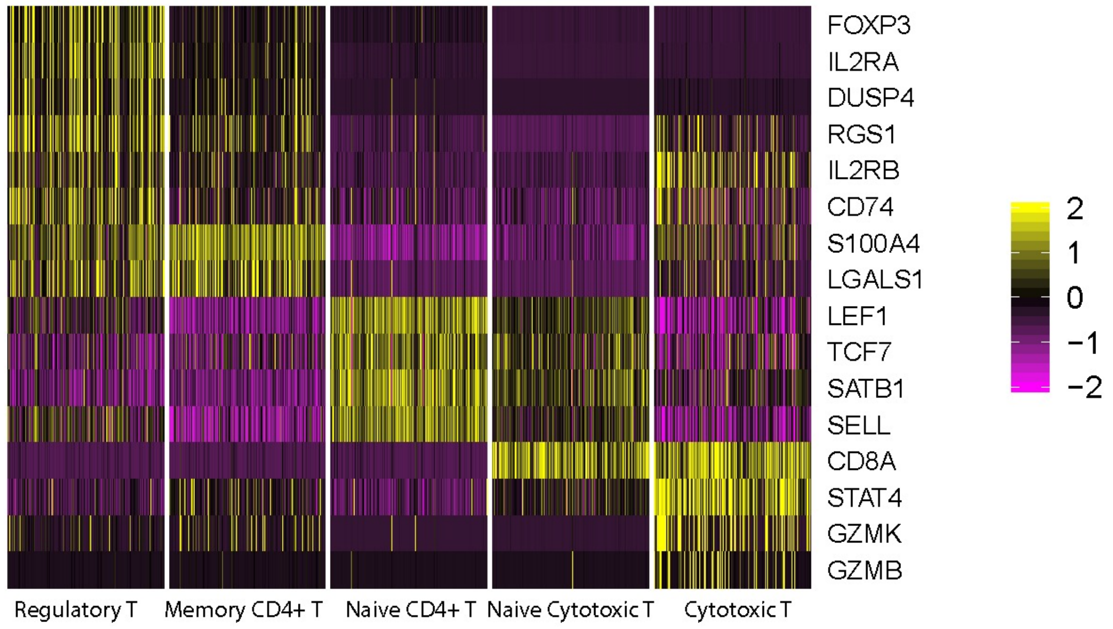
Improve recovering gene expression patterns

Identify the marker genes in each cell type

Original



After Denoising



MAGIC (Dijk et. al., Cell 2018)

- Use cell-cell similarity to improve data quality

- Core idea
    - Calculate cell-cell similarity matrix (KNN graph) A
        - scRNA-seq normalization and PCA
        - Gaussian kernel transformation on the Euclidean distance

$$A(i,j) = e^{-\left(\frac{Dist(i,j)}{\sigma}\right)^2}$$

            - σ is actually cell dependent like tSNE $\sigma(i) = distance(i,\ neighbor(i, ka))$
        - Only retain k nearest neighbors to retain sparsity of A
        - Make A symmetric and positive definite
    - Covert A into a transition probability matrix M $M(i,j) = \frac{A(i,j)}{\sum_k A(i,k)}$

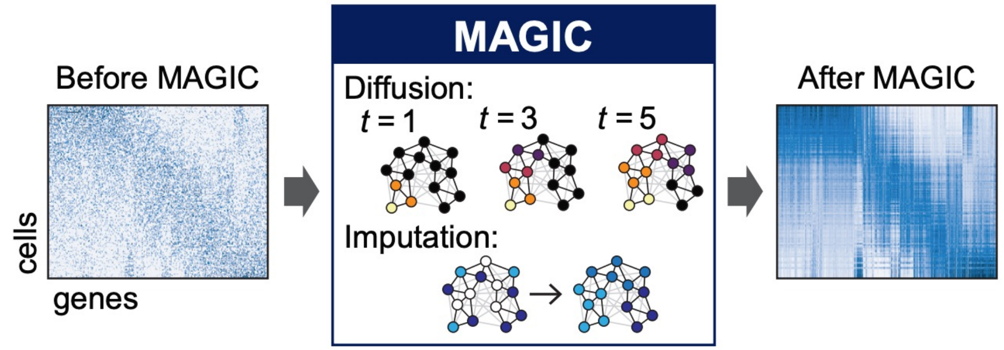
    - Imputation (denoising)

$$D_{imputed}(i,j) = \sum_{k=1}^{n} M^t(i,k) * D(k,j)$$

        - t: Estimated diffusion time

MAGIC (Dijk et. al., Cell 2018)

- Understanding M^t (Diffusion maps, Coifman and Lafon, Appl. Comput. Harmon. Anal., 2006)
  - Small eigenvalues in M can be due to technical noise, M^t reduces the importance of noise dimensions, down-weighting spurious cell neighbors
  - From the perspective of diffusion maps
    - $M^t(i,j)$ represents transition probability from i to j in t steps
    - The authors argued that the first few steps remove noise, while signals will be removed for larger t



- Find the optimal t
  - For each t, calculate

$$\text{R-sq}(\text{data\_t},\ \text{data\_(t-1)}) = 1\text{-SSE}(\text{data\_t},\ \text{data\_(t-1)})/\text{SST}(\text{data\_t},\ \text{data\_(t-1)})$$

  - Choose the smallest t where Rsq is small enough
- This may over-smooth the data

SAVER (Huang et. al., Nature Methods 2018)

- Use gene-gene dependence to improve data quality
- Core idea
  - Assume the data distribution

$$Y_{gc} \sim Poisson\left(s_c \lambda_{gc}\right)$$
$$\lambda_{gc} \sim Gamma\left(\alpha_{gc}, \beta_{gc}\right)$$

  - Use Poisson regression to build a prediction model of one gene on all other genes
    - Add Lasso penalty to increase prediction accuracy
    - More principled to use NB regression, but here the purpose is prediction, use Poisson to reduce computational cost

$$\log\, E\left(Y_{gc}/s_c | Y_{g'c}\right) = \log\, \mu_{gc} = \gamma_{g0} + \sum_{g' \neq g} \gamma_{gg'} \log\left[\frac{Y_{g'c}+1}{s_c}\right]$$

    - Use μ_{gc} as denoised value can over-smooth the data, predict λ_{gc} to faithfully recover true biological randomness of the data

SAVER (Huang et. al., Nature Methods 2018)

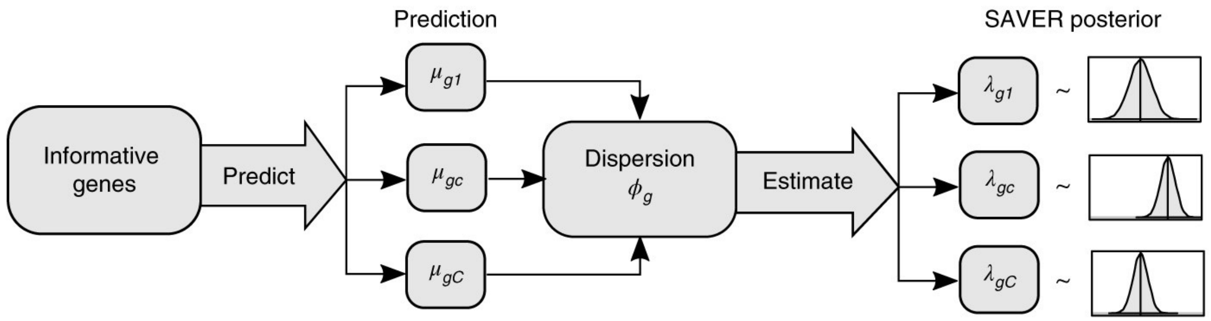
- Use gene-gene dependence to improve data quality

- Core idea
  - Use μ_{gc} as denoised value can over-smooth the data, predict λ_{gc} to faithfully recover true biological randomness of the data

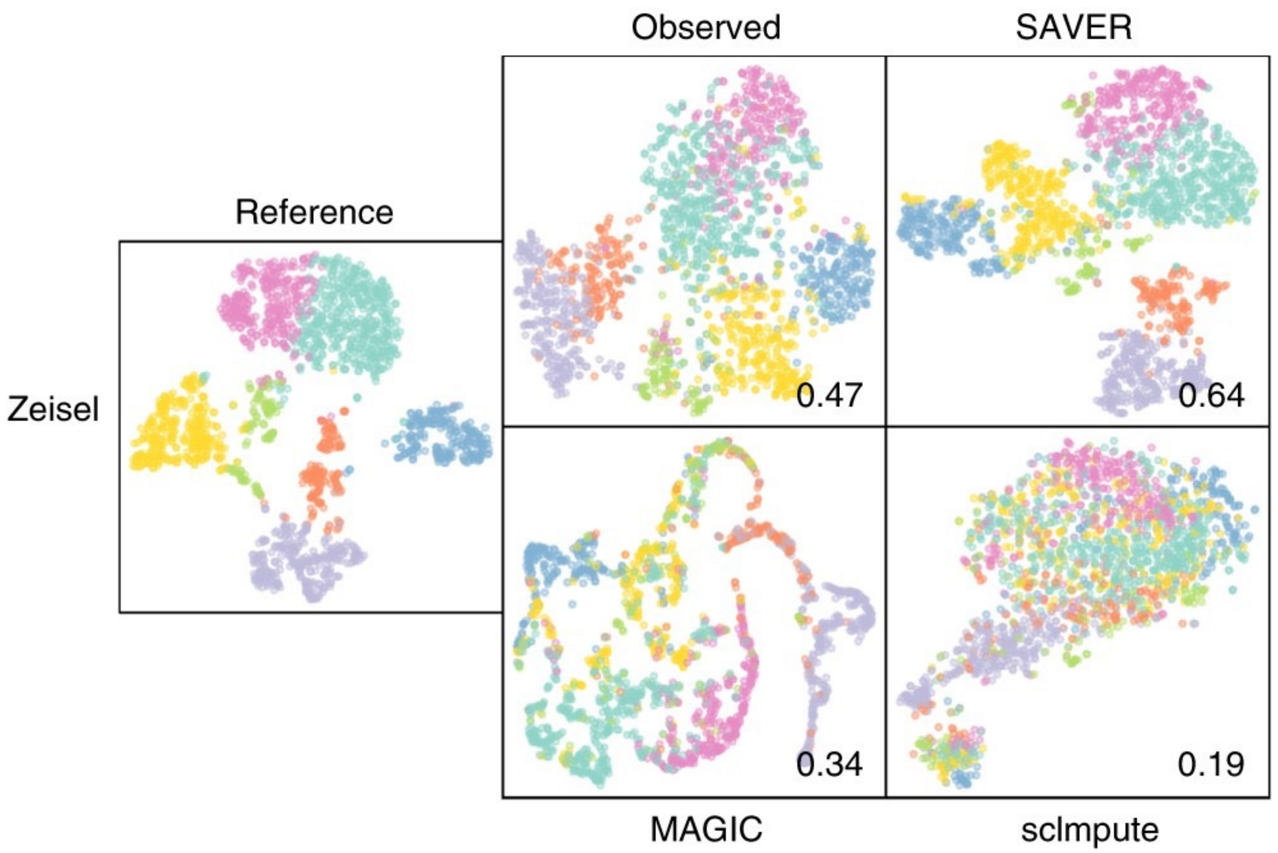
$$Y_{gc} \sim Poisson\left(s_c \lambda_{gc}\right)$$
$$\lambda_{gc} \sim Gamma\left(\alpha_{gc}, \beta_{gc}\right)$$

$$\lambda_{gc} | Y_{gc}, \hat{\alpha}_{gc}, \hat{\beta}_{gc} \sim Gamma\left(Y_{gc} + \hat{\alpha}_{gc}, s_c + \hat{\beta}_{gc}\right)$$

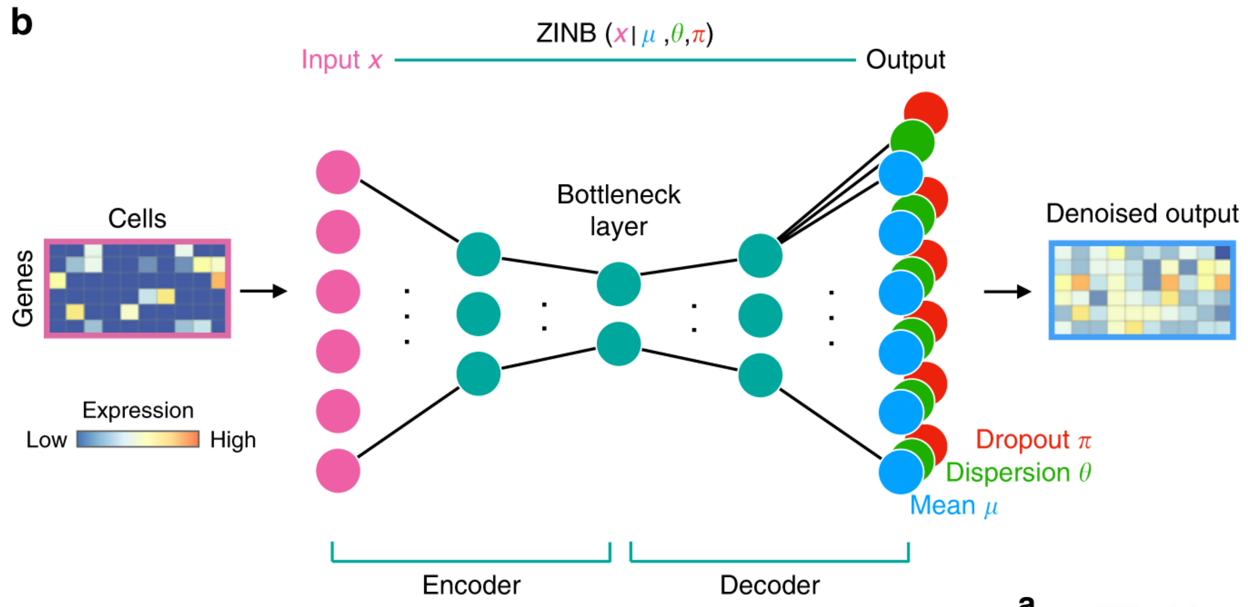
  - Empirical Bayes estimate of the variance parameter
    - Maximize marginal likelihood of three models: Constant variance / dispersion / Fano factor
    - Pick the model that has the largest maximal variance



SAVER (Huang et. al., Nature Methods 2018)



DCA (Eraslan et. al., Nature Communications 2019)

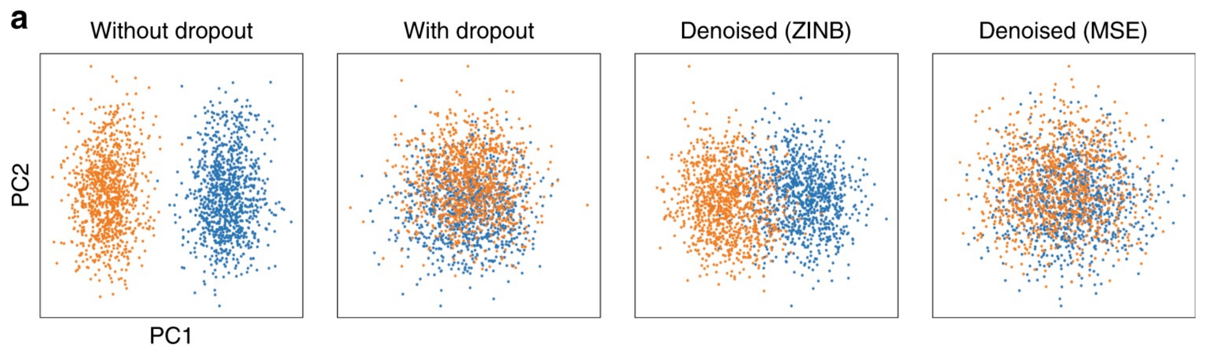


$$\cancel{\log X = UV^T}$$

$$X = g \circ f(X) \qquad U = f(X)$$

Neural Networks

- Use an autoencoder (non-linear factor model)
- Use the ZINB / NB negative log-likelihood as the loss function when training the autoencoder



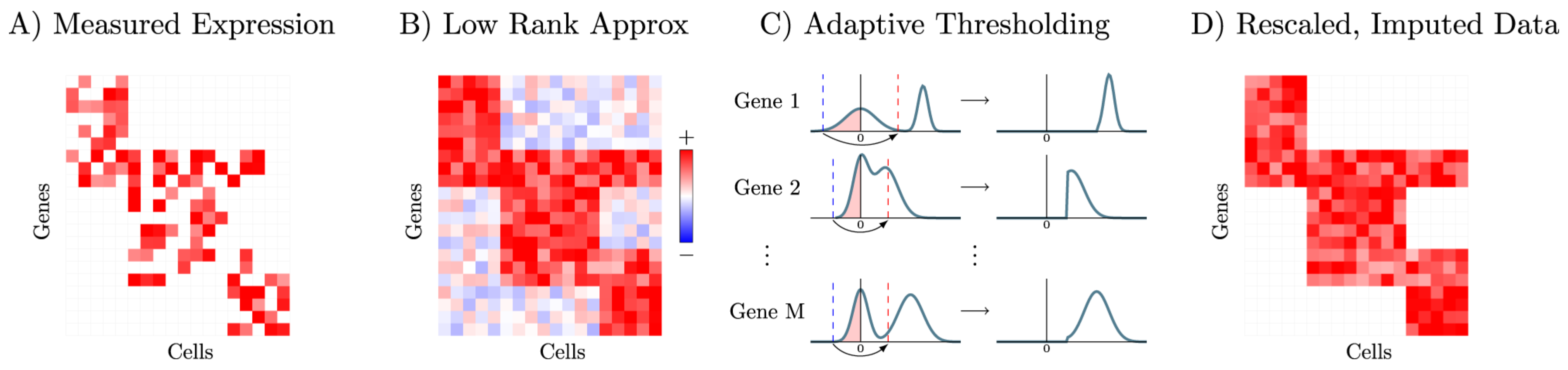
- Similar methods
  - scVI (Lopez et. al., 2018): use variational autoencoder + batch effect correction
  - SAVER-X (Wang et. al., 2019): pretrain the autoencoder on other datasets to borrow information + preserve biological randomness as in SAVER

ALRA (Linderman et. al., Nature Communications 2022)

- Simply uses a linear factor model for matrix denoising

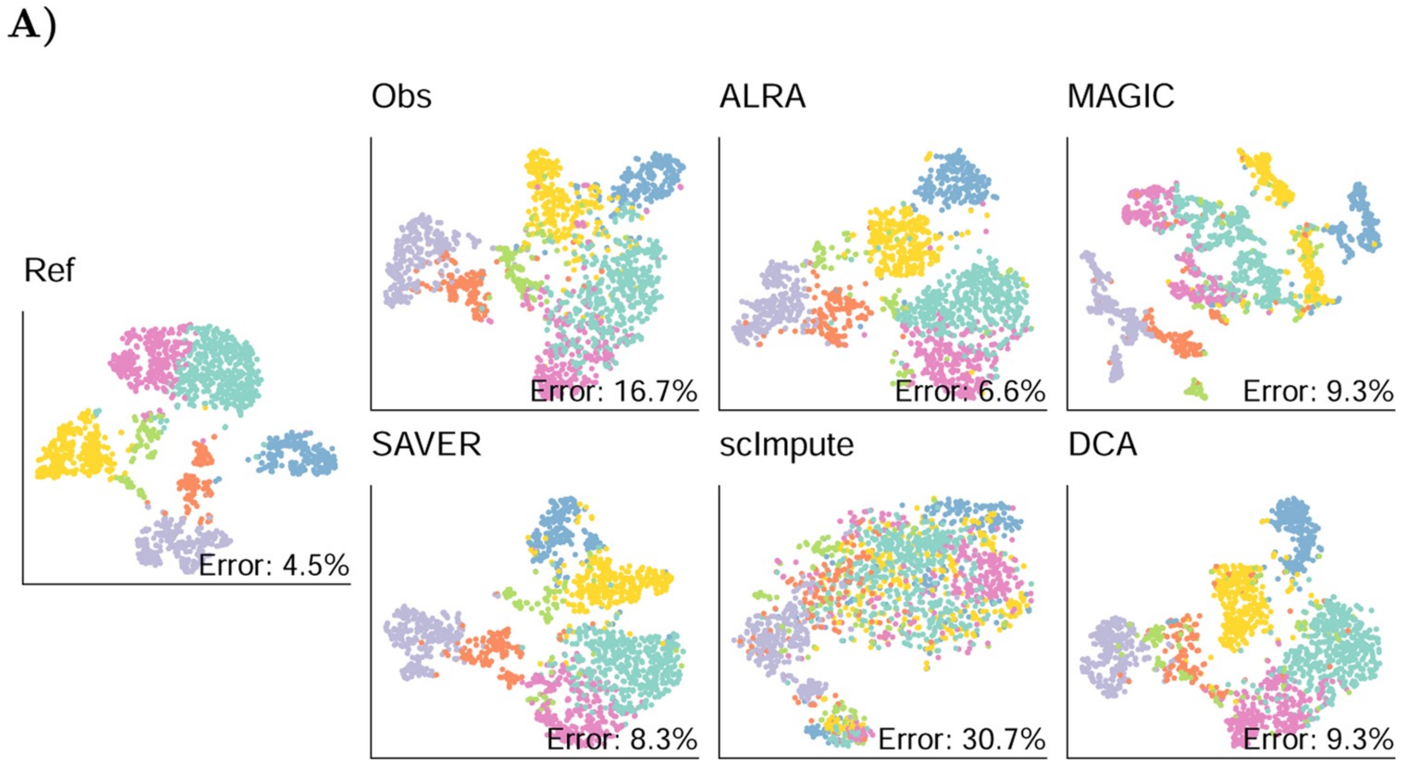
$$\tilde{X} = X + E \qquad X = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

- Assume that the “true” gene expression matrix (signal matrix) is low-rank and sparse



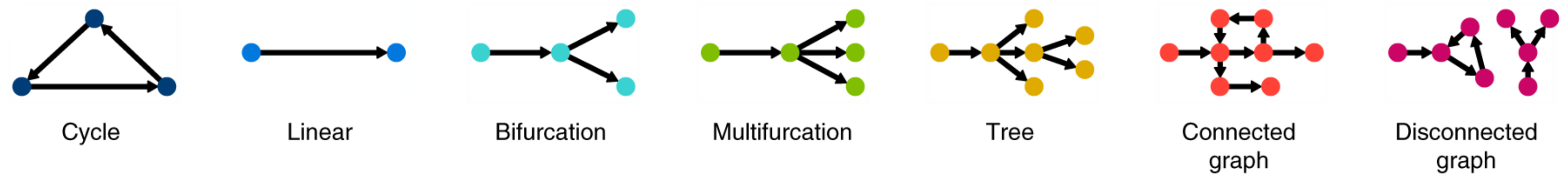
- Idea for preserving the zeros: estimated value of the true zeros in SVD should have a symmetric distribution around 0.
  - Also implicitly assume that nonzero values are large enough

ALRA (Linderman et. al., Nature Communications 2022)



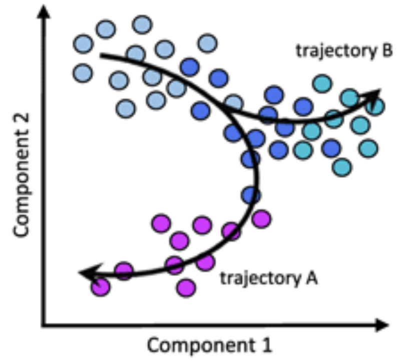
Trajectory inference (TI) for scRNA-seq

- Understand the cell fate decisions in biological processes, such as differentiation, immune response, or cancer expansion with scRNA-seq data
- Infer or assume a type of underlying trajectory structure



Saelens W. et. al., *Nat. Biotech.* **37**, 547–554(2019)

- Computationally project and order the cells along the trajectory



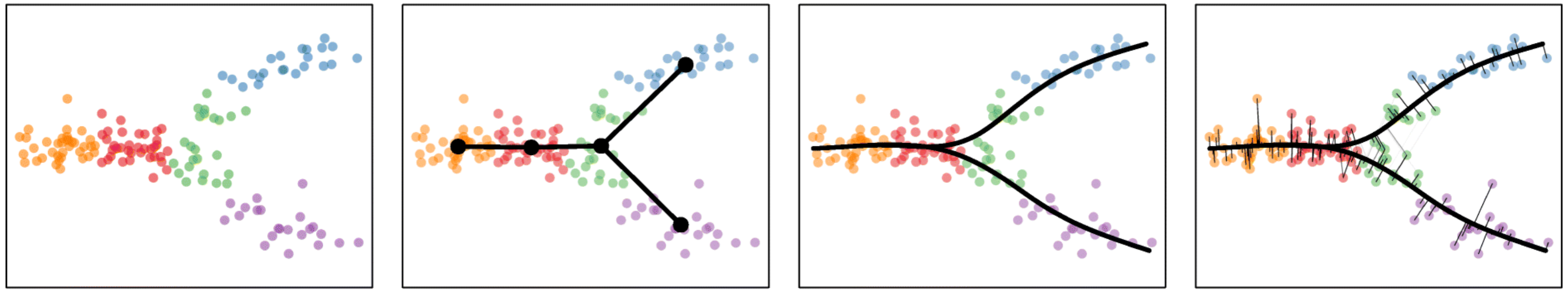
Liu S. *F1000Research* 5 (2016)

- The orders of the cells are also called the pseudotimes

- There already exists more than 70 TI methods
  (For a comprehensive benchmarking, see Saelens W. et. al., *Nat. Biotech.* **37**, 547–554(2019))

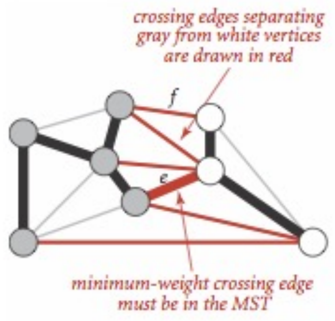
Slingshot (Street et. al., BMC Genomics, 2018)

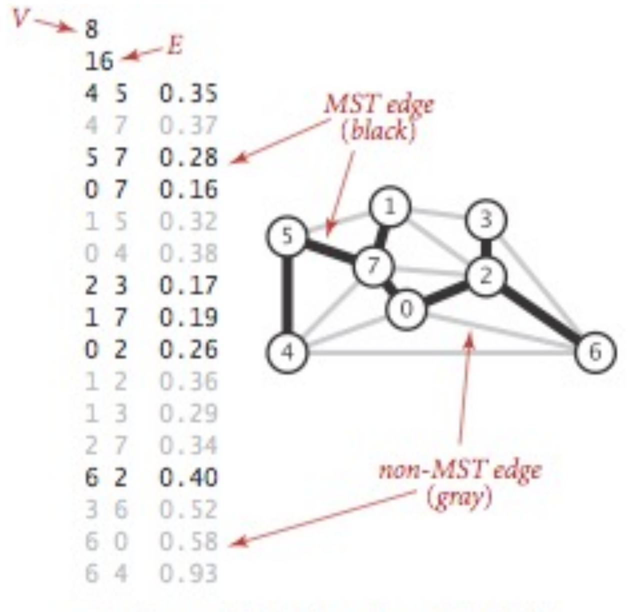
- Idea: build a connection graph for the clusters



- Main steps:
  - Dimension reduction and clustering
  - Treat clusters as nodes in a graph and draw a minimum spanning tree (MST)
    - MST: spanning tree whose weights (sum of its edge weights) is the smallest among spanning trees
    - Cut property: Given any cut in an edge-weighted graph (with all edge weights distinct), the crossing edge of minimum weight is in the MST of the graph.
    - Tutorial: https://algs4.cs.princeton.edu/43mst/
    - Edge weight: distance between two clusters

$$d^2(\mathcal{C}_i, \mathcal{C}_j) \equiv (\bar{X}_i - \bar{X}_j)^T (S_i + S_j)^{-1} (\bar{X}_i - \bar{X}_j)$$


crossing edges separating gray from white vertices are drawn in red
f
e
minimum-weight crossing edge must be in the MST



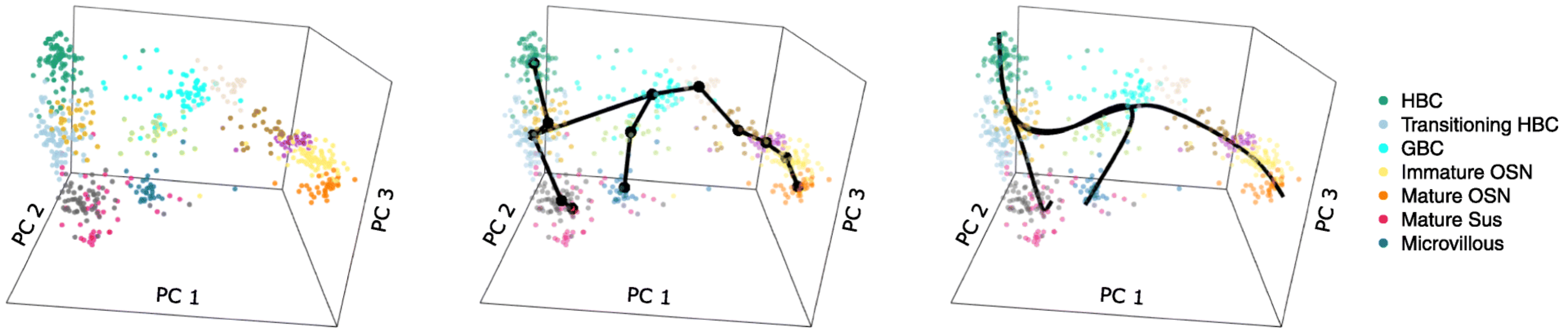
8
16

4	5	0.35
4	7	0.37
5	7	0.28
0	7	0.16
1	5	0.32
0	4	0.38
2	3	0.17
1	7	0.19
0	2	0.26
1	2	0.36
1	3	0.29
2	7	0.34
6	2	0.40
3	6	0.52
6	0	0.58
6	4	0.93

An edge-weighted graph and its MST

Slingshot (Street et. al., BMC Genomics, 2018)

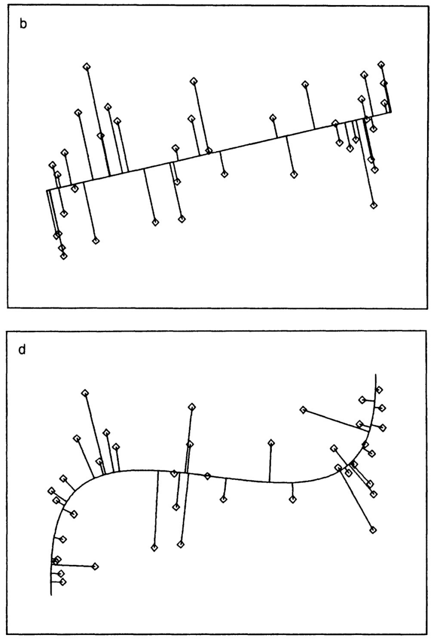
- Main steps:
  - Estimate the lineage (trajectory) structure
    - Dimension reduction and clustering
    - Treat clusters as nodes in a graph and draw a minimum spanning tree (MST)
    - Undirected tree -> directed tree: user provided initial cluster
      - Perform constrained MST if users provide the leaf node
  - Drawback: what if the lineage structure is not a tree?
  - Estimate a cell pseudotime
    - For each lineage (path from initial node to a leaf node), fit a principal curve and project the cells onto the principal curve to determine the pseudotime



- Challenge: shared lineages should have overlapping principal curves and cells belonging to multiple lineages should have similar pseudotime estimates

Slingshot (Street et. al., BMC Genomics, 2018)

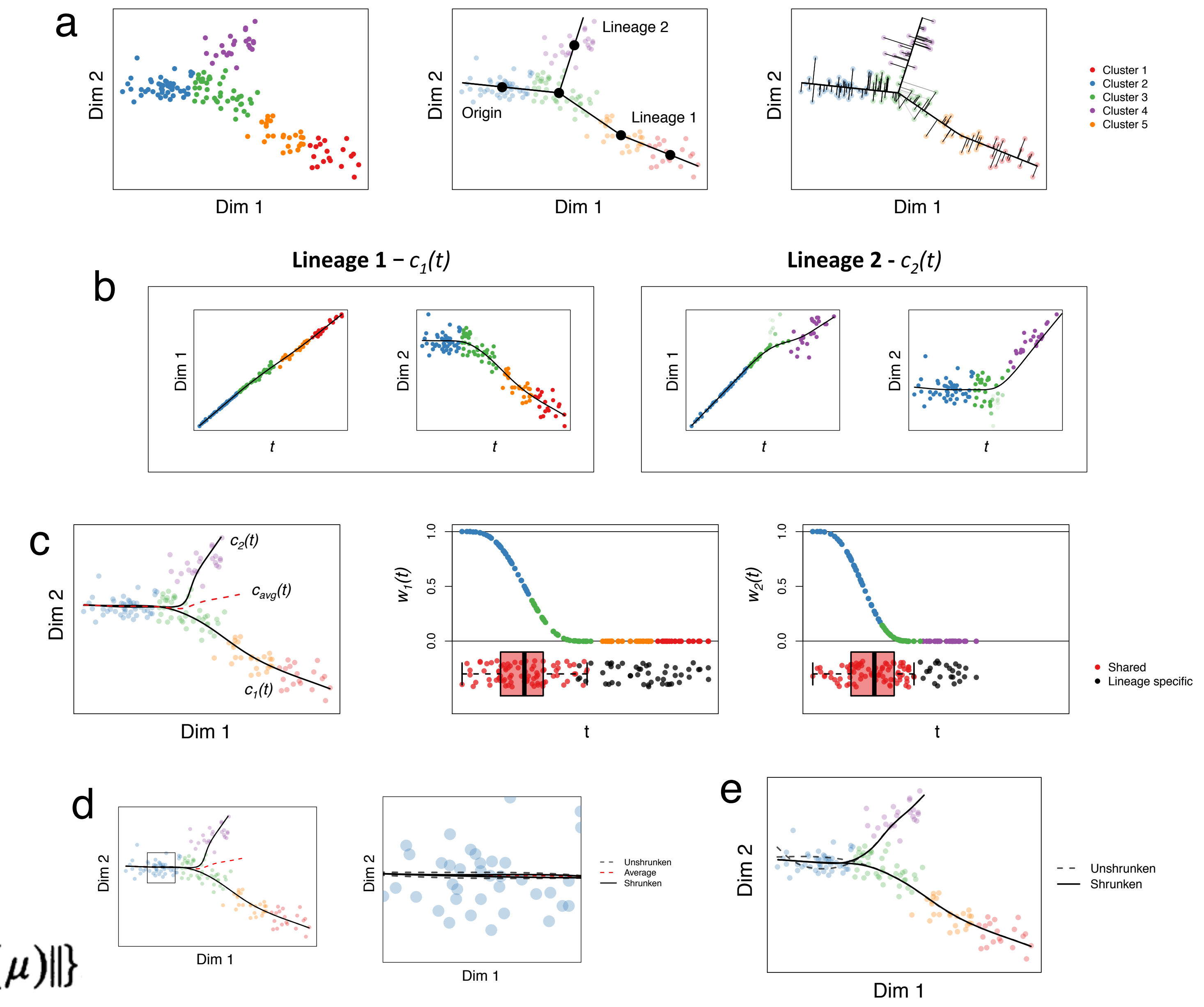
- Principal curve (Hastie and Stuetzle, JASA 1989)



- Generalization of getting first (linear) PC

$$\mathbf{x}_i = \mathbf{f}(\lambda_i) + \mathbf{e}_i$$

$$\lambda_{\mathbf{f}}(\mathbf{x}) = \sup_{\lambda}\{\lambda : \|\mathbf{x} - \mathbf{f}(\lambda)\| = \inf_{\mu}\|\mathbf{x} - \mathbf{f}(\mu)\|\}$$



PAGA (Wolf et. al., Genome Biology, 2019)

- Construct KNN graph of the data (use any reasonable method, can apply denoising first)
- Clustering and determine connectivity between clusters based on the KNN graph
  - ε_{ij}^{sym}: number of edges (outgoing and ingoing) between cluster i and j
  - Under the “null” where there is no connection between the two clusters

$$p_{\mathrm{arbit}}(\varepsilon | e_i, e_j, n_i, n_j, n) \simeq \mathcal{N}(\varepsilon | \hat{\varepsilon}^{\mathrm{sym}}(e_i, e_j, n_i, n_j, n), \hat{\sigma}^{\mathrm{sym}}(e_i, e_j, n_i, n_j, n))$$

$$\text{with} \quad \hat{\varepsilon}^{\mathrm{sym}}(e_i, e_j, n_i, n_j, n) = \frac{e_i n_j + e_j n_i}{n-1},$$

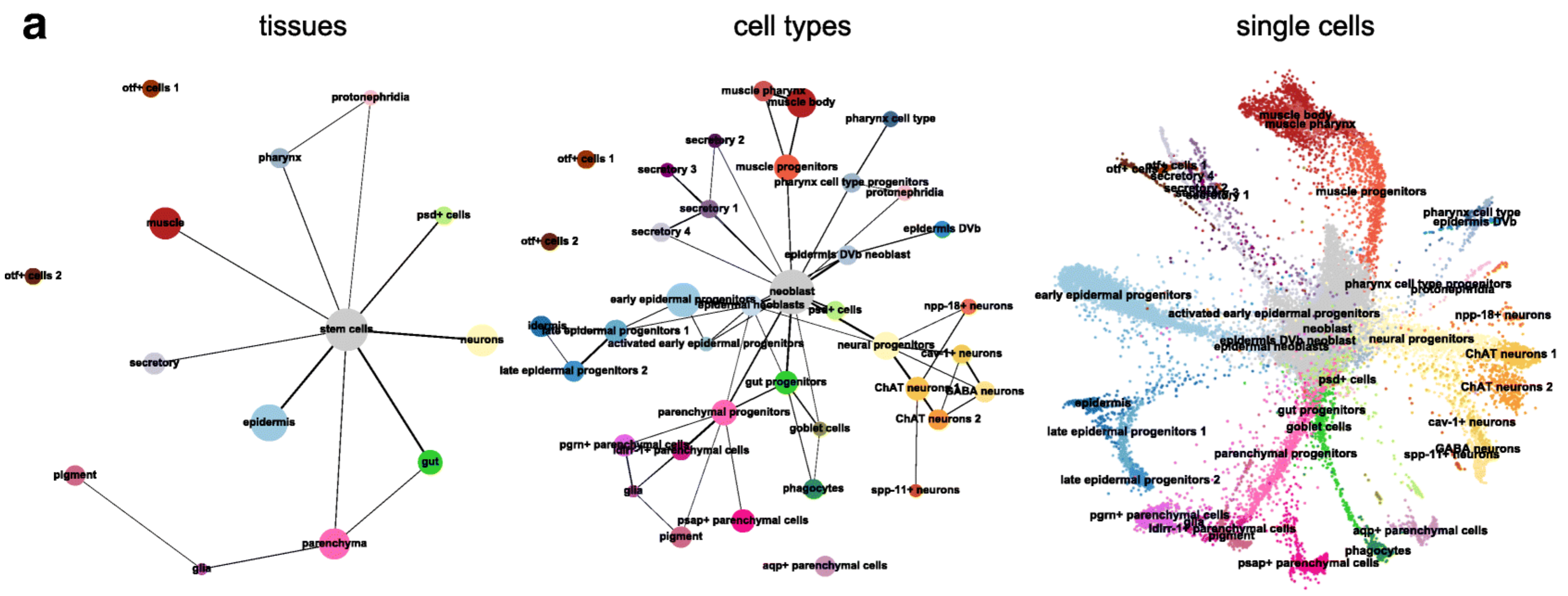
$$\hat{\sigma}^{\mathrm{sym}}(e_i, e_j, n_i, n_j, n) = \frac{e_i n_j (n - n_j - 1) + e_j n_i (n - n_i - 1)}{(n-1)^2}.$$

    - n_i: number of nodes in cluster i, e_i: number of outgoing edges of cluster i
  - Cluster connectivity score:

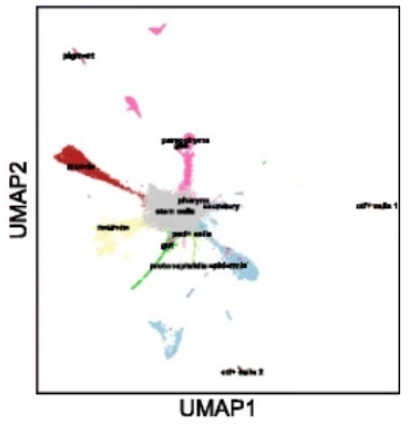
$$c_{ij} = \begin{cases} \frac{\varepsilon_{ij}^{\mathrm{sym}}}{\hat{\varepsilon}^{\mathrm{sym}}(e_i, e_j, n_i, n_j, n)} & \text{if } \varepsilon_{ij}^{\mathrm{sym}} < \hat{\varepsilon}^{\mathrm{sym}}(e_i, e_j, n_i, n_j, n) \\ 1 & \text{else.} \end{cases}$$

    - The paper discussed that “equivalently”, if each cluster has a Gaussian density, cluster connectivity score reflects overlapping region of the density functions
  - Thresholding cluster connectivity score to get the final trajectory structure

PAGA (Wolf et. al., Genome Biology, 2019)



- Initialize UMAP with the coarse cluster graph leads to better visualization of the data



PAGA (Wolf et. al., Genome Biology, 2019)

- Construct KNN graph of the data (use any reasonable method, can apply denoising first)
- Clustering and determine connectivity between clusters based on the KNN graph
- Pseudotime estimation for each cell
  - Pseudotime defined as the distance of a continuous progression along a manifold
  - Based on a diffusion maps model on the cell-cell graph (like MAGIC, cell-cell transition matrix T)
  - Some highlights of the algorithm
    - Laplace transformation

$$\widetilde{L} = I - \widetilde{T}, \quad \widetilde{T} = D^{\frac{1}{2}} T D^{-\frac{1}{2}}$$

    - Calculate diffusion pseudotime based on the eigenvectors and eigenvalues of L (or equivalently, T)

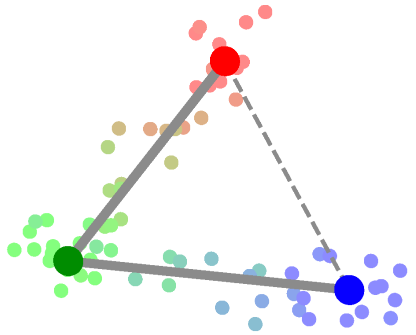
$$\widetilde{\mathrm{dpt}}^2(\iota_1, \iota_2) = \sum_{r=2}^{n_{\mathrm{nodes}}} \left(\frac{\lambda_r}{1-\lambda_r}\right)^2 (\widetilde{v}_{r\iota_1} - \widetilde{v}_{r\iota_2})^2$$

    - Making using of trajectory structure: assign ∞ to cell-cell distance for cells in disconnected clusters

$$\widetilde{\mathrm{dpt}}(\iota_1, \iota_2) = \sum_{r=n_{\mathrm{comps}}+1}^{n_{\mathrm{nodes}}} \left(\frac{\lambda_r}{1-\lambda_r}\right)^2 (\widetilde{v}_{r\iota_1} - \widetilde{v}_{r\iota_2})^2 + \sum_{r=1}^{n_{\mathrm{comps}}} (\widetilde{v}_{r\iota_1} - \widetilde{v}_{r\iota_2})^2$$

VITAE (Du et. al., BioRXiv, 2023)

- Combine a graph-based method and direct modeling of the data using variational autoencoder
- Assume a complete graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$
    - $\mathcal{N}(\mathcal{G})$: a vertex denotes a distinct cell state / type
    - $\mathcal{E}(\mathcal{G})$: an edge denotes a possible transition between two cell states/types
- A cell position $\tilde{\boldsymbol{w}}_i \in [0,1]^k$ on the graph



$$\tilde{\boldsymbol{w}}_i = \begin{cases} \boldsymbol{e}_j & \text{if cell } i \text{ is on vertex } j \in \{1, \cdots, k\} \\ w_i \boldsymbol{e}_{j_1} + (1 - w_i)\boldsymbol{e}_{j_2} & \text{if cell } i \text{ is on the edge between vertices } j_1 \text{ and } j_2 \ (j_1 \neq j_2) \end{cases}$$

- The trajectory backbone, $\mathcal{B}$, as a subgraph of $\mathcal{G}$

$$\mathcal{N}(\mathcal{B}) = \mathcal{N}(\mathcal{G}) \qquad \mathcal{E}(\mathcal{B}) = \left\{ (j_1, j_2) \in \mathcal{E}(\mathcal{G}) : \sum_i \mathbb{1}_{\{\tilde{w}_{ij_1} > 0, \tilde{w}_{ij_2} > 0\}} > 0 \right\}$$

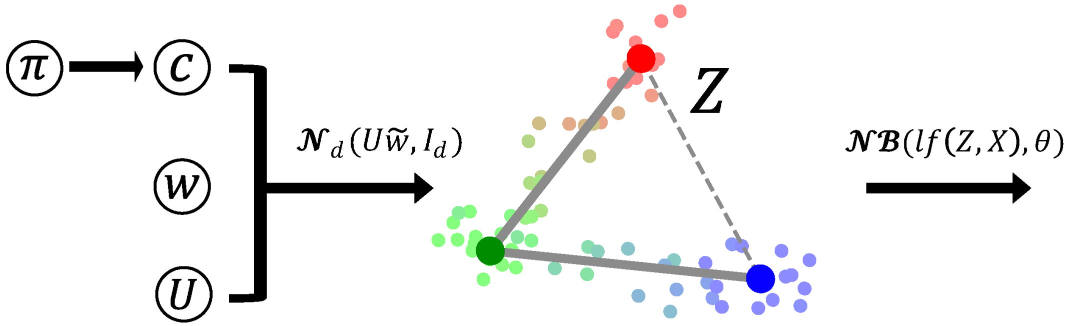
VITAE (Du et. al., BioRXiv, 2023)

$$w_i \overset{\text{i.i.d.}}{\sim} \text{Uniform}(0,1)$$

$$c_i \overset{\text{i.i.d.}}{\sim} \text{Multinomial}(1, \boldsymbol{\pi}),$$

$$w_i \perp\!\!\!\perp c_i$$

$$\tilde{\boldsymbol{w}}_i = w_i \boldsymbol{a}_{c_i} + (1 - w_i)\boldsymbol{b}_{c_i}$$



π
c
w
U
$\mathcal{N}_d(U\tilde{w}, I_d)$
Z
$\mathcal{NB}(lf(Z,X),\theta)$

Observed count data Y_{ig}

	Gene$_1$	Gene$_2$	Gene$_3$...
Cell$_1$	1	2	4
Cell$_2$	0	1	0
Cell$_3$	1	1	3
Cell$_4$	0	2	1
Cell$_5$	1	1	5
Cell$_6$	3	0	1
Cell$_7$	2	0	2
...			...

- Assume latent variables $\boldsymbol{Z}_i \in \mathbb{R}^d$ satisfy

$$\boldsymbol{Z}_i | \tilde{\boldsymbol{w}}_i \sim \mathcal{N}_d(\boldsymbol{U}\tilde{\boldsymbol{w}}_i, \boldsymbol{I}_d)$$

A non-linear mapping from the latent space to the high-dimensional observed data

Model f_g by a neural network

  - $\boldsymbol{U}$: unknown positions of the vertices in $\mathbb{R}^d$
  - $\boldsymbol{X}_i$: cell-specific confounding covariates (data source, cell cycle, et. al.)

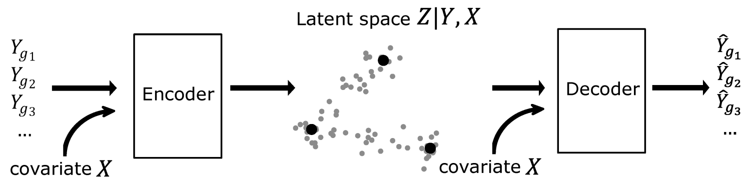
- We also assume a mixture prior on $\tilde{\boldsymbol{w}}_i$

VITAE (Du et. al., BioRXiv, 2023)

- Key contribution: Simultaneous batch effect removal and trajectory analysis
- Loss function:

Reconstruction loss

$$
\begin{aligned}
L = &-(1-\alpha)\sum_{i=1}^{N}\mathbb{E}_{q(\boldsymbol{Z}_i|\mathbf{Y}_i,\boldsymbol{X}_i)}\log p(\mathbf{Y}_i|\boldsymbol{Z}_i,\boldsymbol{X}_i) \\
&+\beta\sum_{i=1}^{N}D_{\mathrm{KL}}(q(\boldsymbol{Z}_i|\mathbf{Y}_i,\boldsymbol{X}_i)\|p(\boldsymbol{Z}_i)) \\
&-\alpha\sum_{i=1}^{N}\log p(\mathbf{Y}_i|\boldsymbol{Z}_i=\mathbf{0}_d,\boldsymbol{X}_i) \\
&+\kappa\,\Omega_{\mathrm{MMD}}(\mathcal{D}_N) \\
&+\gamma\,\Omega_{\mathrm{Jacobian}}(\mathcal{D}_N).
\end{aligned}
$$



- Four penalty terms:
  - β-VAE:
    - Set $\beta > 1$ to encourage posteriors of $\boldsymbol{Z}_i$ to lie along trajectory backbone
  - Adjust for confounding $\boldsymbol{X}_i$ and batch effects
    - Soft penalty: help decorrelate $\boldsymbol{Z}_i$ from $\boldsymbol{X}_i$
    - MMD loss: used across replicates where the cell populations are known to be the same
  - Jacobian regularizer
    - enhance stability in optimization

$$
\Omega_{\mathrm{Jacobian}}(\mathcal{D}_N)=\sum_{i=1}^{N}\sum_{j=1}^{d}\sum_{g=1}^{G}\mathbb{E}_{q(\boldsymbol{Z}_i|\mathbf{Y}_i,\boldsymbol{X}_i)}\left[\left(\frac{\partial \boldsymbol{Z}_{ij}}{\partial \mathbf{Y}_{ig}}\right)^2\right]
$$

GPfates (Lonnberg et. al., Science Immunology, 2017)

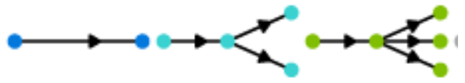
- Model (normalized and dimension-reduced) scRNA-seq data as generated from a mixture of Gaussian processes

$$X = f_c(t) + \varepsilon \quad p(F|T) = \prod_{c=1}^{C} \mathcal{N}(f_c|0, \boldsymbol{K}_t^c)$$

$$k(t_{n_1}, t_{n_2}) = \sigma_{\mathrm{SE}}^2 \exp\left(-\frac{|t_{n_1} - t_{n_2}|^2}{2l_{\mathrm{SE}}^2}\right)$$

  - Infer posterior $t|X$ to estimate each cell's pseudotime
  - Prior distribution $p(t_n) = \mathcal{N}(\mathrm{day}_n, \sigma_{\mathrm{prior}}^2)$
    - Make use of the calendar time

  - Use variational Bayes and EM to infer parameters
  - For interpretation of each GP component, only allow one branching point



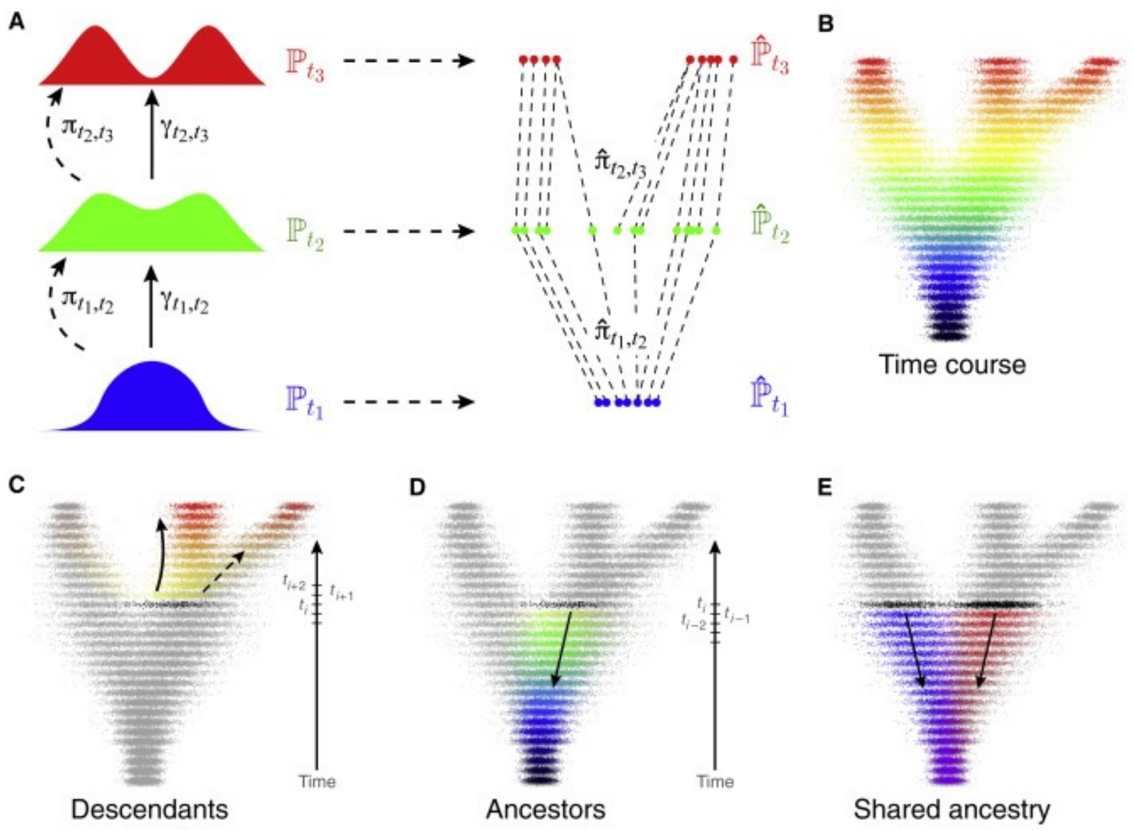
Waddington-OT (Schiebinger et. al., Cell, 2019)

- Make use the cell collection time and assume that cells having a later collection time are descendants of the earlier collected cells
- Estimate transition between cells instead of pseudotime
  - Optimal transport coupling

$$
\begin{aligned}
\pi_{s,t}(\epsilon) = \underset{\pi}{\text{minimize}} \quad & \iint c(x,y)\pi(x,y)dxdy - \epsilon \iint \pi(x,y)\log \pi(x,y)dxdy \\
\text{subject to} \quad & \int \pi(x,\cdot)dx = \mathbb{Q}_s \\
& \int \pi(\cdot,y)dy = \mathbb{P}_t.
\end{aligned}
$$

  - Corresponding optimization problem

$$
\begin{aligned}
\hat{\pi}_{t_i,t_{i+1}} \quad = \quad \underset{\pi}{\arg\min} \quad & \sum_{x\in S_i}\sum_{y\in S_{i+1}} c(x,y)\pi(x,y) - \epsilon \iint \pi(x,y)\log\pi(x,y)dxdy \\
& + \lambda_1 \mathrm{KL}\left[\sum_{x\in S_i}\pi(x,y)\middle\| d\hat{\mathbb{P}}_{t_{i+1}}(y)\right] + \lambda_2 \mathrm{KL}\left[\sum_{y\in S_{i+1}}\pi(x,y)\middle\| d\hat{\mathbb{Q}}_{t_i}(x)\right]
\end{aligned}
$$



Related papers

- Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., ... & Pe’er, D. (2018). Recovering gene interactions from single-cell data using data diffusion. *Cell*, *174*(3), 716-729.
- Huang, M., Wang, J., Torre, E., Dueck, H., Shaffer, S., Bonasio, R., ... & Zhang, N. R. (2018). SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods*, *15*(7), 539-542.
- Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., & Theis, F. J. (2019). Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications*, *10*(1), 390.
- Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., & Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature methods*, *15*(12), 1053-1058.
- Wang, J., Agarwal, D., Huang, M., Hu, G., Zhou, Z., Ye, C., & Zhang, N. R. (2019). Data denoising with transfer learning in single-cell transcriptomics. *Nature methods*, *16*(9), 875-878.
- Linderman, G. C., Zhao, J., Roulis, M., Bielecki, P., Flavell, R. A., Nadler, B., & Kluger, Y. (2022). Zero-preserving imputation of single-cell RNA-seq data. *Nature communications*, *13*(1), 192.

- Street, K., Risso, D., Fletcher, R. B., Das, D., Ngai, J., Yosef, N., ... & Dudoit, S. (2018). Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC genomics*, *19*, 1-16.
- Wolf, F. A., Hamey, F. K., Plass, M., Solana, J., Dahlin, J. S., Göttgens, B., ... & Theis, F. J. (2019). PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome biology*, *20*, 1-9.
- Du, J. H., Chen, T., Gao, M., & Wang, J. (2023). Model-based trajectory inference for single-cell rna sequencing using deep learning with a mixture prior. *bioRxiv*, 2020-12.
- Lönnberg, T., Svensson, V., James, K. R., Fernandez-Ruiz, D., Sebina, I., Montandon, R., ... & Teichmann, S. A. (2017). Single-cell RNA-seq and computational analysis using temporal mixture modeling resolves TH1/TFH fate bifurcation in malaria. *Science immunology*, *2*(9), eaal2192.
- Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., ... & Lander, E. S. (2019). Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, *176*(4), 928-943.

- Coifman, R. R., & Lafon, S. (2006). Diffusion maps. *Applied and computational harmonic analysis*, *21*(1), 5-30.
- Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American statistical association*, *84*(406), 502-516.