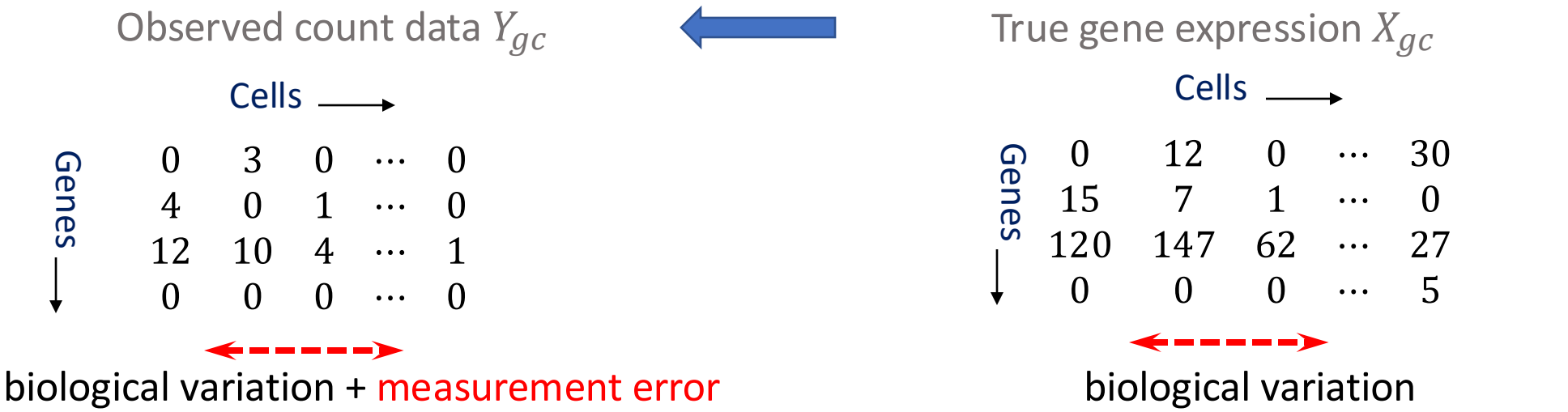
Lecture 6
scRNA-seq denoising

Outline

- scRNA-seq denoising methods
  - MAGIC
  - SAVER
  - DCA
  - ALRA
  - More methods to discuss later: scVI (batch correction) / SAVER-X (transfer learning)

scRNA-seq denoising

- scRNA-seq data is very noisy



Data denoising: get an estimate of X

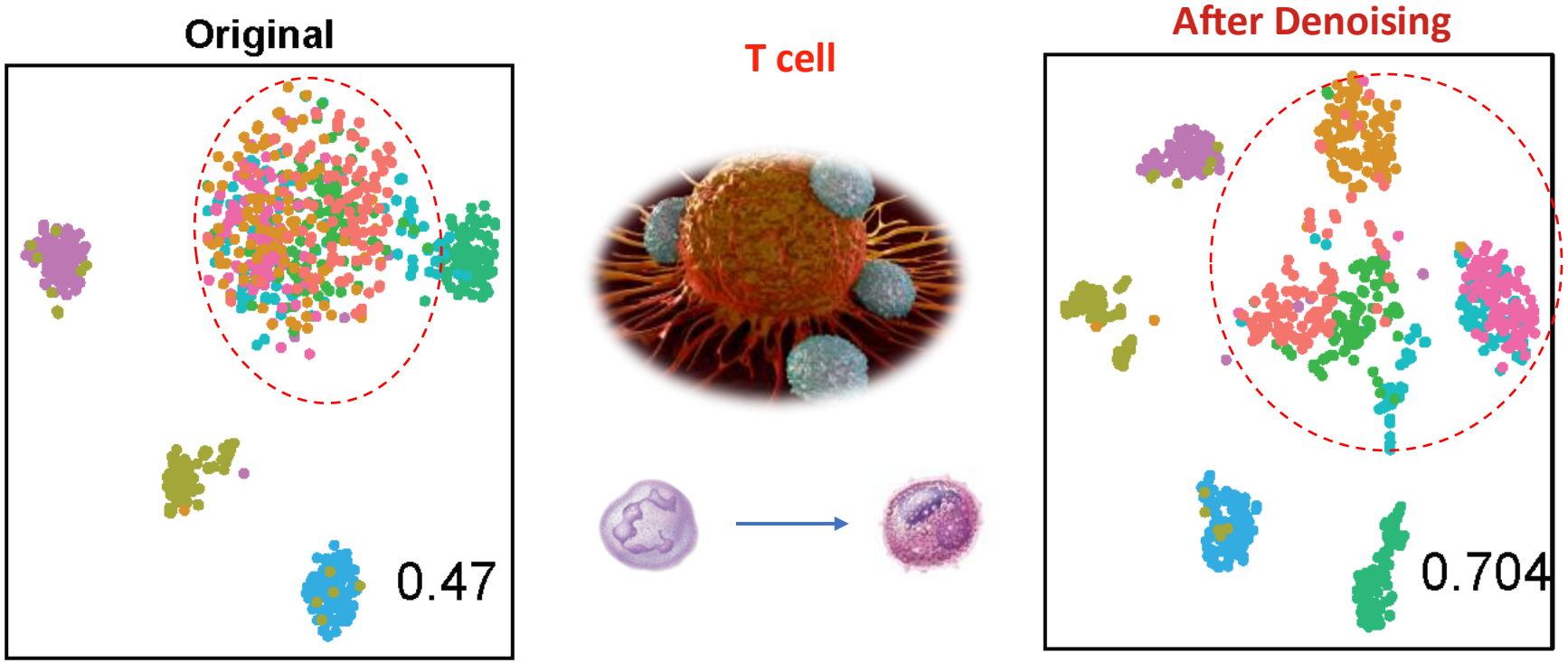
Core idea:

- 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?

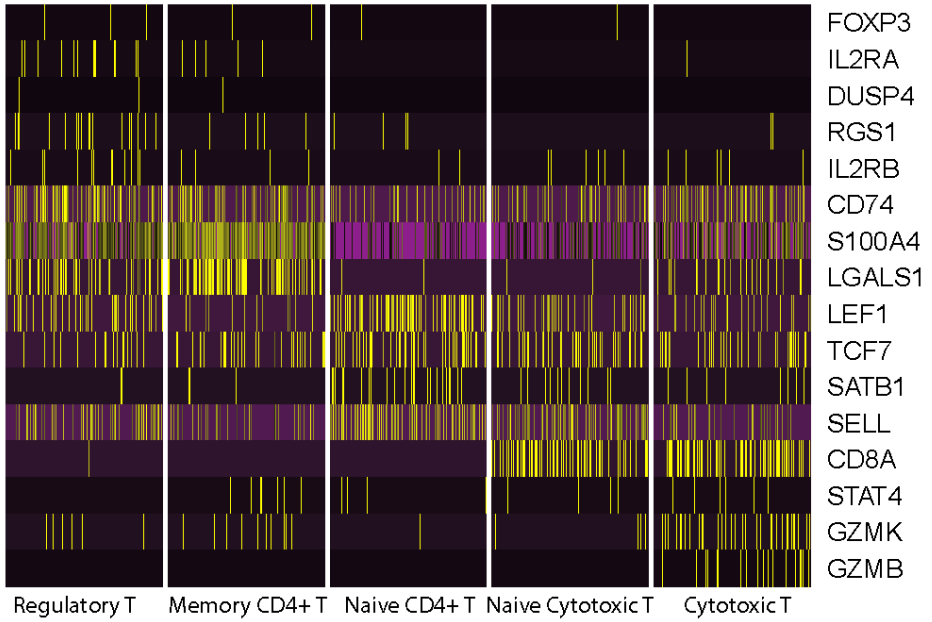
900 PBMC cells (immune cells in peripheral blood) with labels [Zheng et. al., 2017]



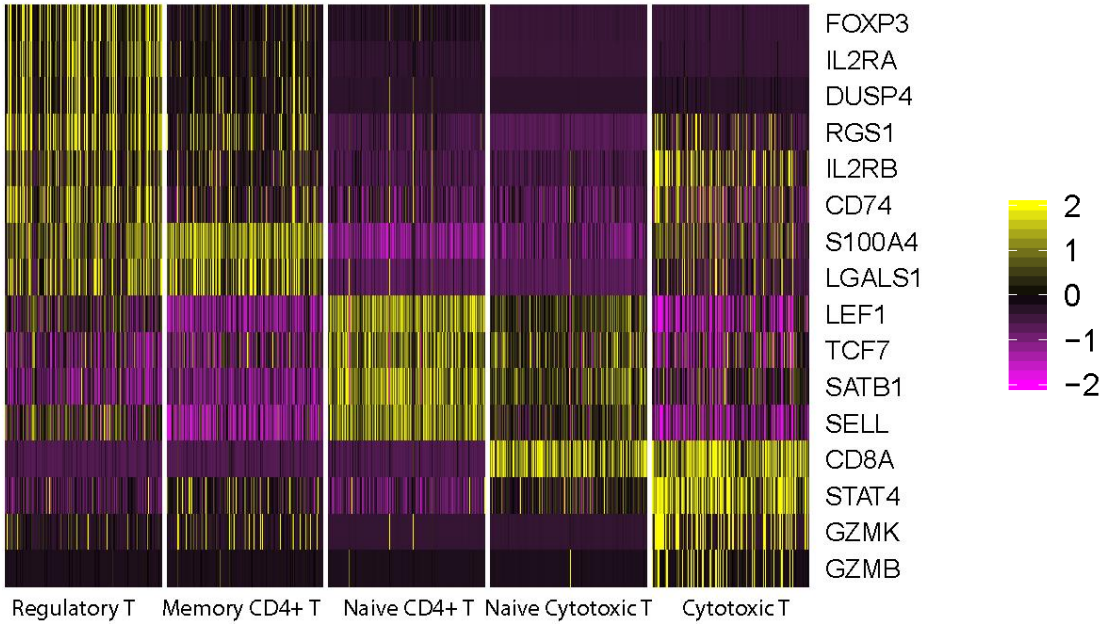
Improve recovering gene expression patterns

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

$$M(i,j) = \frac{A(i,j)}{\sum_k A(i,k)}$$

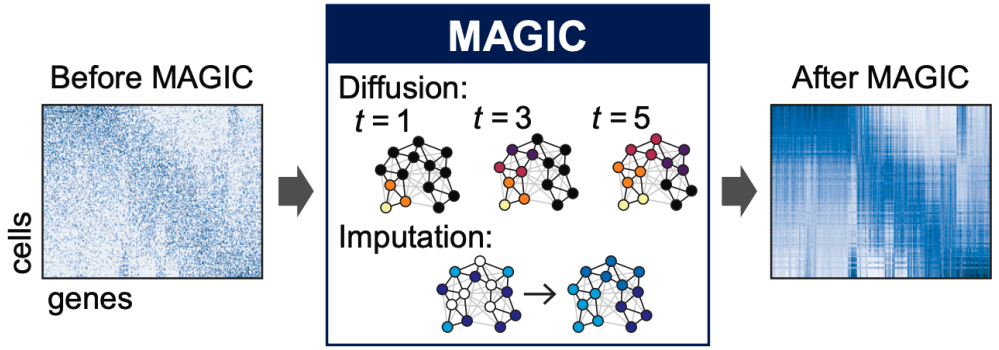
  - Imputation (denoising)

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

    - 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)

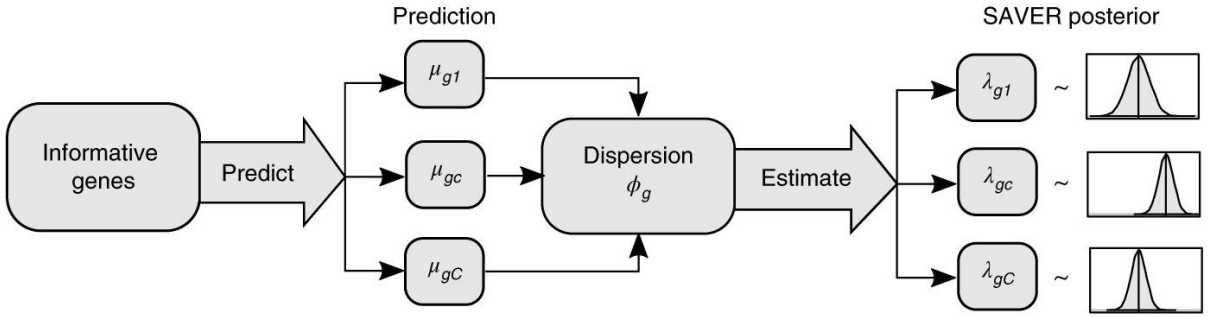
- Use gene-gene dependence to improve data quality

- 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

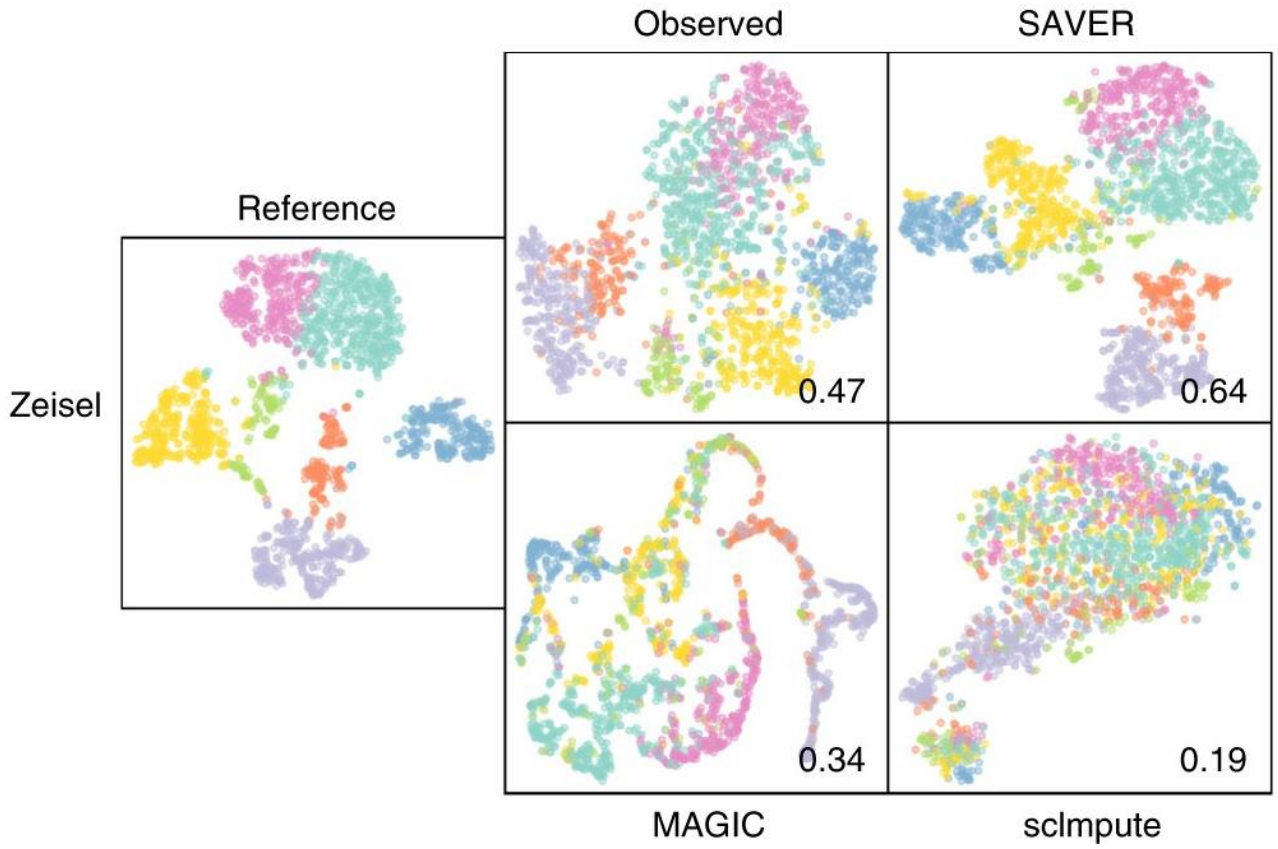
$$Y_{gc} \sim Poisson\left(s_c \lambda_{gc}\right)$$
$$\lambda_{gc} \sim Gamma\left(\alpha_{gc}, \beta_{gc}\right)$$

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

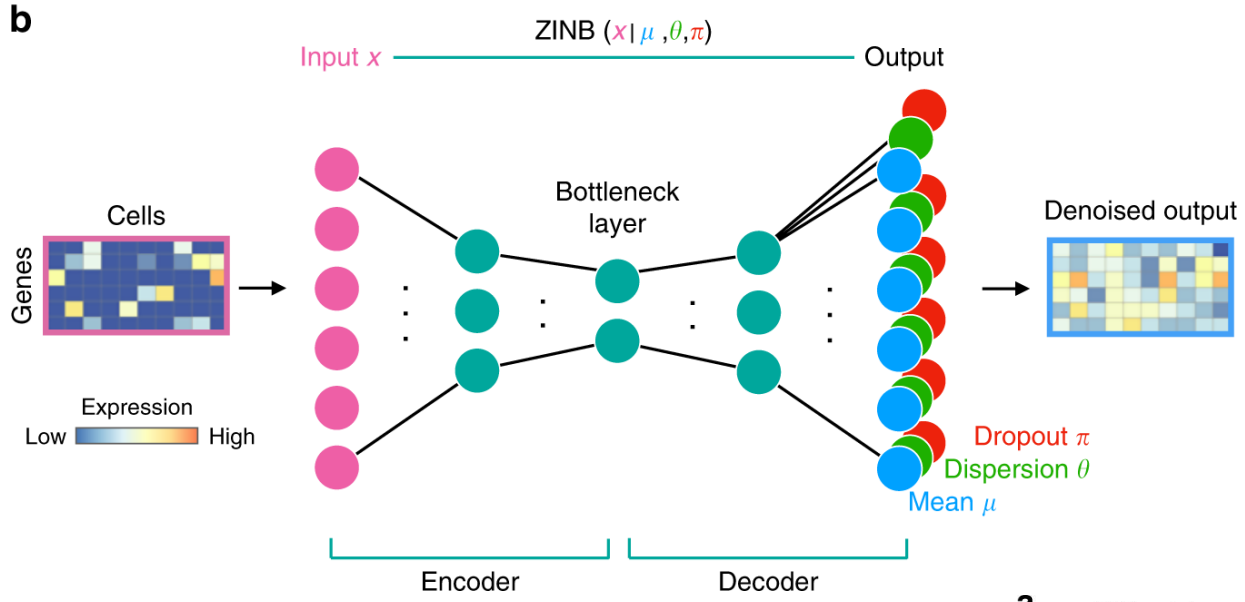
  - Empirical Bayes estimate of the variance parameter
    - Maximize marginal likelihood of three models: Constant variance / dispersion / Fano factor
    - Pick the model that has the largest maximal variance



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



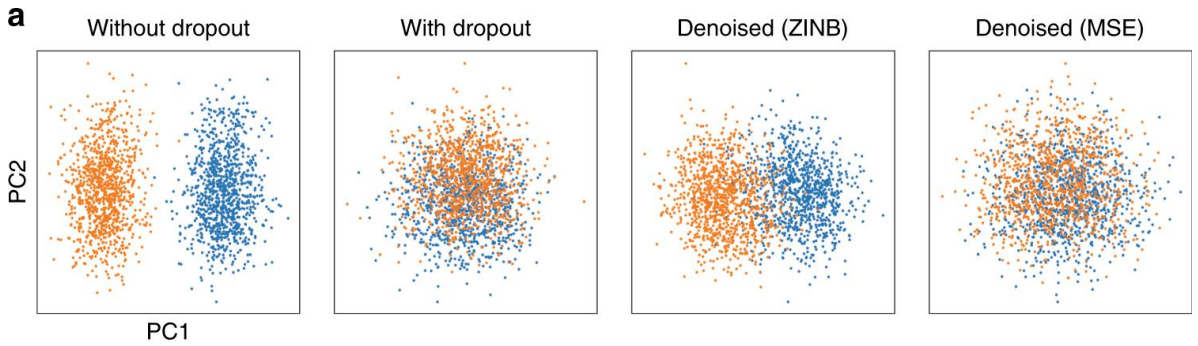
DCA (Eraslan et. al., Nature Communications 2019)



$$\log X = UV^T$$

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

Neural Networks



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

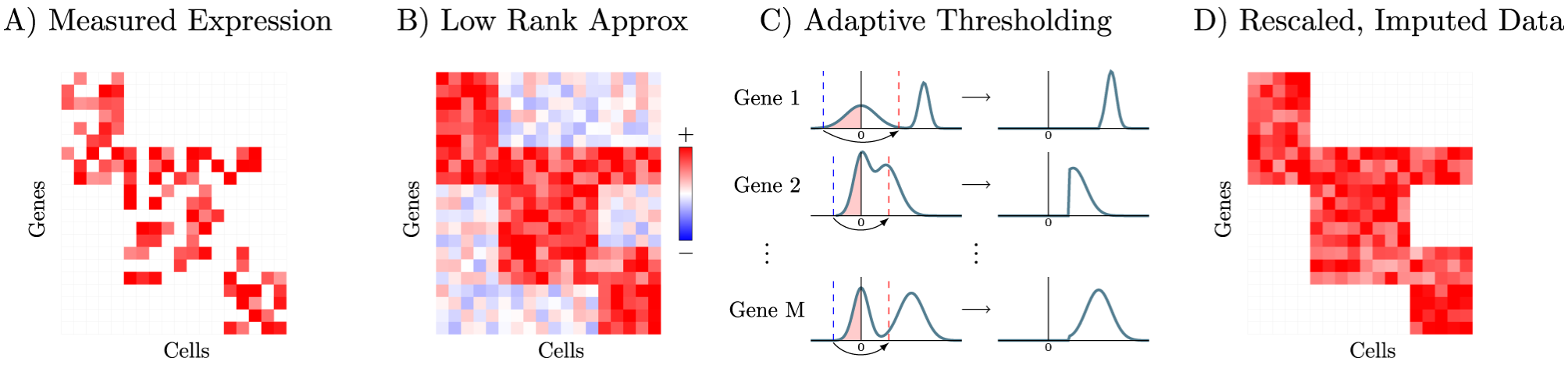
- Similar methods
  - scVI (Lopez et. al., 2018): use variational autoencoder + batch effect correction
  - SAVER-X (Wang et. al., 2019): pretrain the autoencoder on other datasets to borrow information + preserve biological randomness as in SAVER

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

- Simply uses a linear factor model for matrix denoising

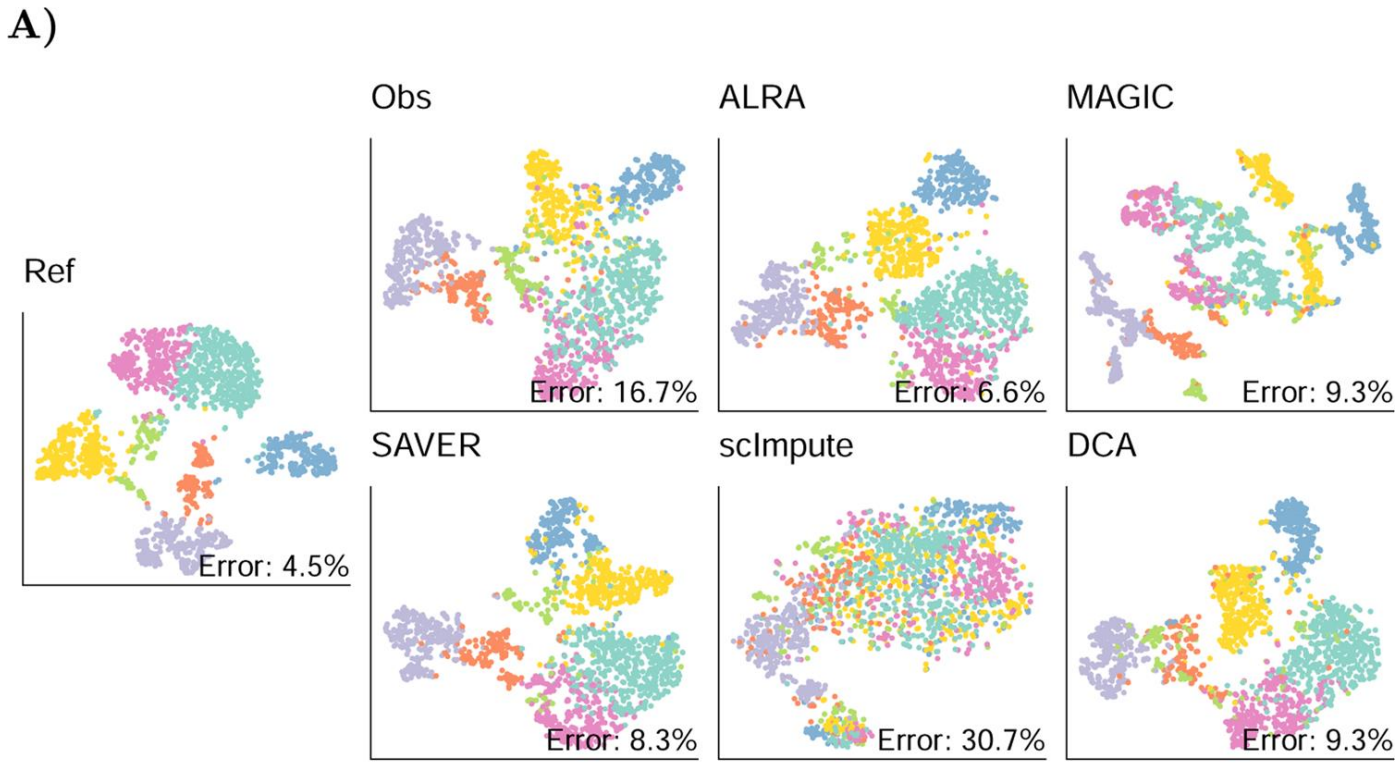
$$\tilde{X} = X + E \qquad X = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

- 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)



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.