STAT 35510
Lecture 7

Spring, 2024

Jingshu Wang

Outline

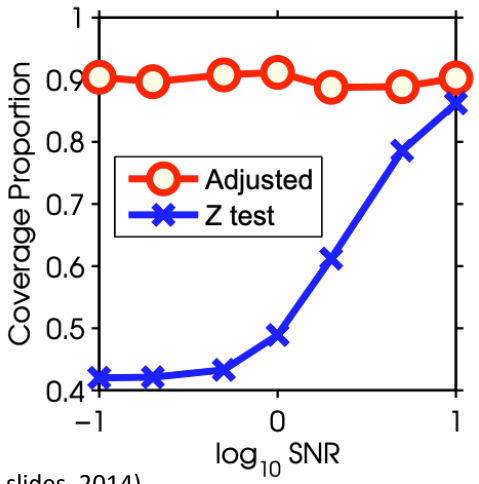
- “Post-estimation” inference in scRNA-seq
  - Hypotheses testing after clustering
    - Conditional tests
    - Data thinning
    - Simulate global null data
  - Hypotheses testing after trajectory inference
  - Hypotheses testing and gene property estimation after denoising

Post selection bias in linear regression

- In linear regression, we may want to select a smaller model if number of covariates is too large
- A naïve procedure for linear regression inference with model selection
  - Perform a variable selection procedure: stepwise with AIC/BIC, lasso, elastic net, …
  - Fit linear regression (OLS) only using the selected covariates
  - Construct 95% confidence intervals $(\hat{\beta}_j - 1.96\hat{\sigma}_j, \hat{\beta}_j + 1.96\hat{\sigma}_j)$
  - Test the hypothesis $H_0: \beta_j = 0$ by rejecting when $|\hat{\beta}_j / \hat{\sigma}_j| \geq 1.96$

- These confidence intervals are invalid if model selection and inference in performed on the same dataset

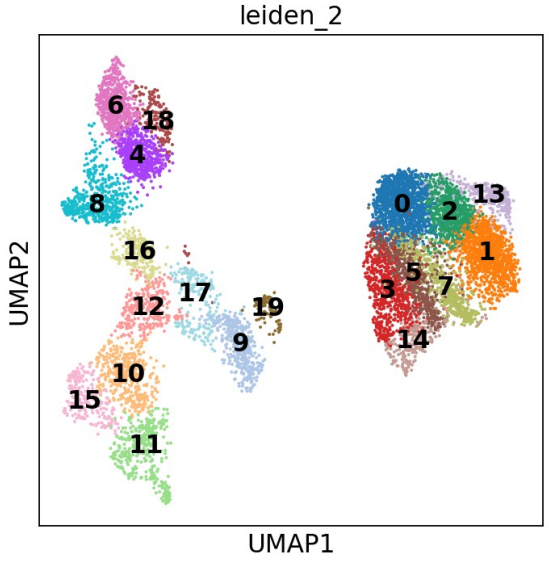
- A possible solution is sample splitting:
  split the data into two, one for model selection,
  one for testing / constructing CI

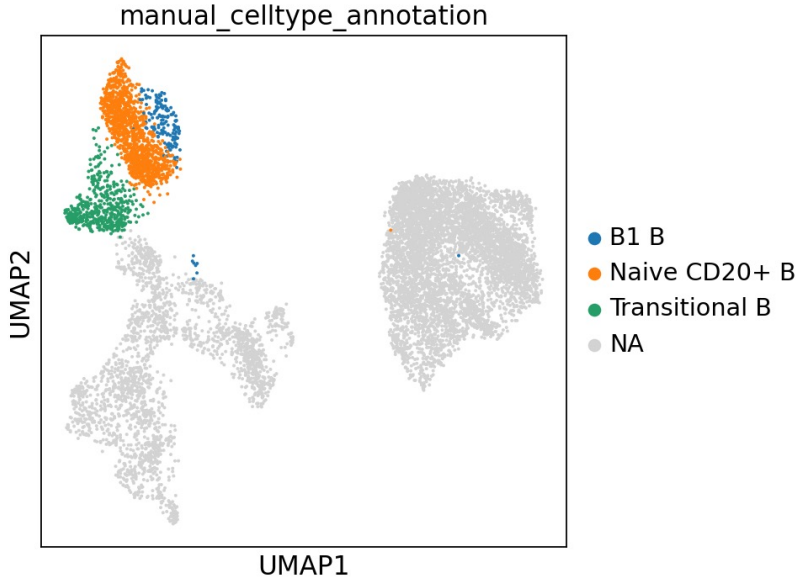


Large signal so that model selection has no error (selected model is not random)

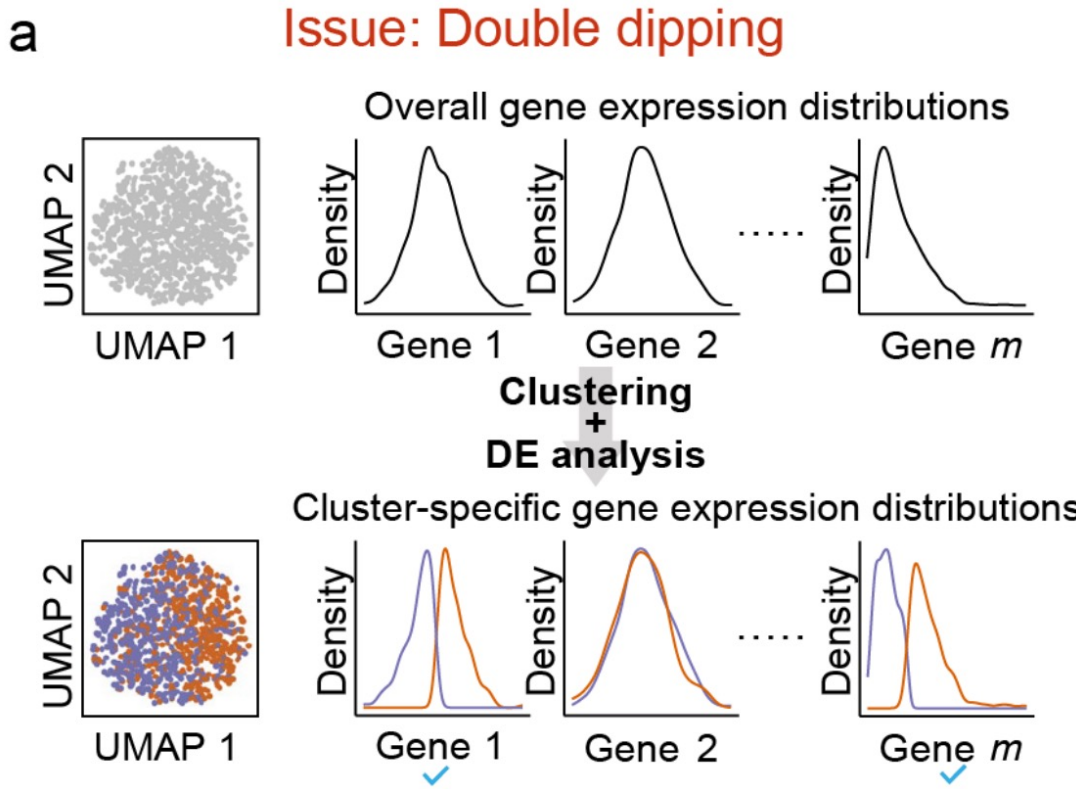
(Figure from Jason Lee slides, 2014)

Bias in post clustering differential testing



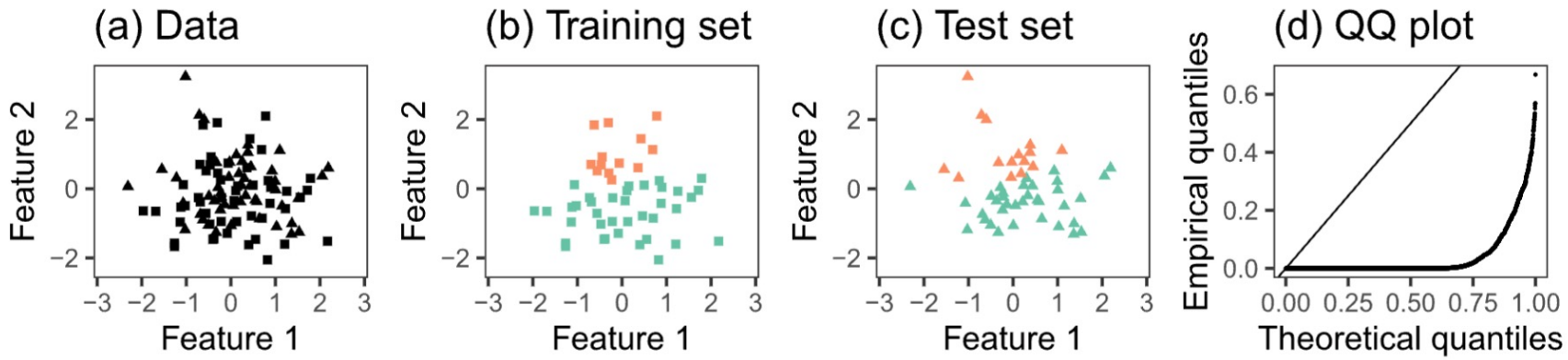


- Differential gene expression testing can have many false positives if clusters are not separated well
- Consequence: identified marker genes not replicable across samples



Bias in post clustering differential testing

- True cell label for a cell i Z_i, gene expression level for a gene g Y_{ig}
  - Idea null hypothesis: $H_{0g}: Z_i \perp Y_{ig}$
  - Challenge: Z_i is not observed, we can only obtain an estimate $\hat{Z}_i = \hat{f}(Y_{i\cdot})$
    - Under H_{0g}, $\hat{f}(Y_{i\cdot})$ can still depend on Y_{ig} as $\hat{f}(\cdot)$ is learnt by the data and $\hat{f}(Y_{i\cdot})$ is a function of Y_{ig}
    - Sample splitting would not help in unsupervised learning: sample splitting makes $\hat{f}(\cdot)$ independent from from the data but $\hat{Z}_i$ is still a function of Y_{ig}



Selected inference idea

- Selective inference methods developed by Witten group
  - Assume that the gene expressions (after normalizing) follows multivariate independent normal distributions

$$\mathbf{X} \sim \mathcal{MN}_{n\times q}(\boldsymbol{\mu}, \mathbf{I}_n, \sigma^2\mathbf{I}_q)$$

    - Can be extended to allowing a known covariance matrix Σ across features
    - Allow each cell to have a different mean vector $\boldsymbol{\mu}_i$
  - A clustering algorithm provide a data-dependent partition of the observations

  - For any pair of clusters, test for the null hypothesis whether the average of the mean vectors of two estimated clusters are the same or not

$$H_0^{\{\widehat{C}_1,\widehat{C}_2\}} : \bar{\mu}_{\widehat{C}_1} = \bar{\mu}_{\widehat{C}_2} \quad \text{versus} \quad H_1^{\{\widehat{C}_1,\widehat{C}_2\}} : \bar{\mu}_{\widehat{C}_1} \neq \bar{\mu}_{\widehat{C}_2}$$

  - Drawback: test for the global null: reject the null if any of the genes are differentially expressed, only evaluates whether a split is true or false
    - Maybe used to combine spurious clusters?

Selective inference idea

- Selective inference (high level idea)
  - Reject $H_0^{\{\hat{C}_1,\hat{C}_2\}}$ if $\|\bar{X}_{\widehat{C}_1} - \bar{X}_{\widehat{C}_2}\|_2$ is large enough
  - Need to know its null distribution conditioning on observed clustering result

$$\mathbb{P}_{H_0^{\{\widehat{C}_1,\widehat{C}_2\}}}\left(\|\bar{X}_{\widehat{C}_1} - \bar{X}_{\widehat{C}_2}\|_2 \geq \|\bar{x}_{\widehat{C}_1} - \bar{x}_{\widehat{C}_2}\|_2 \,\middle|\, \widehat{C}_1, \widehat{C}_2 \in \mathcal{C}(\mathbf{X})\right)$$

  - Not possible as the mean vectors of the cells are not fully under $H_0^{\{\hat{C}_1,\hat{C}_2\}}$
  - Need to condition on additional events to make the conditional null distribution of the test statistics trackable

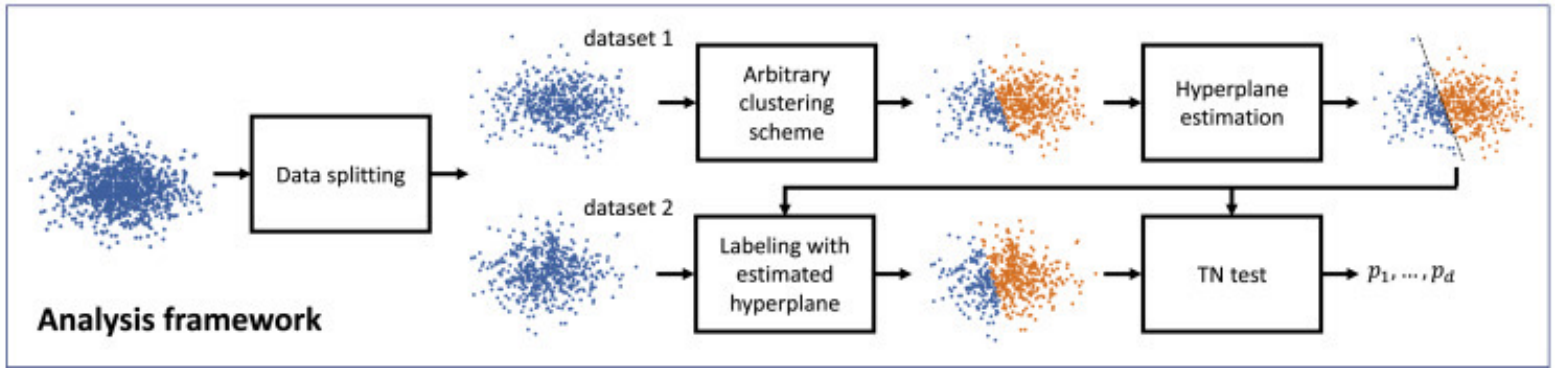
$$p(\mathbf{x}; \{\widehat{C}_1, \widehat{C}_2\}) = \mathbb{P}_{H_0^{\{\widehat{C}_1,\widehat{C}_2\}}}\Big(\|\bar{X}_{\widehat{C}_1} - \bar{X}_{\widehat{C}_2}\|_2 \geq \|\bar{x}_{\widehat{C}_1} - \bar{x}_{\widehat{C}_2}\|_2 \Big| \widehat{C}_1, \widehat{C}_2 \in \mathcal{C}(\mathbf{X}), \boldsymbol{\pi}^{\perp}_{\nu(\widehat{C}_1,\widehat{C}_2)}\mathbf{X} = \boldsymbol{\pi}^{\perp}_{\nu(\widehat{C}_1,\widehat{C}_2)}\mathbf{x}, \operatorname{dir}\left(\bar{X}_{\widehat{C}_1} - \bar{X}_{\widehat{C}_2}\right) = \operatorname{dir}\left(\bar{x}_{\widehat{C}_1} - \bar{x}_{\widehat{C}_2}\right)\Big),$$

  - (Gao et. al. JASA 2022) has shown that the test statistics follow a truncated chi-square distribution
    - The truncation event can be explicitly characterized if clustering algorithm is hierarchical clustering (Gao et. al. JASA 2022) or k-means clustering (Chen and Witten, JMLR 2023)
    - Limitation: requires a clustering algorithm with clear analytical form

TN test (Zhang et. al., Cell Systems 2019)

- Work for “any” clustering algorithm (via approximations + sample splitting)
- Test for one gene at a time allowing for other genes to be truly differentially expressed
- Strong distribution assumptions on the observed gene expressions
  - When testing between two clusters, assume that the observed data comes from a two-component Gaussian mixture
  - Each component represents a cluster label
  - Assume independence across genes (like the selective inference idea, should allow a known covariance matrix Σ across features)

- Incorporate the data splitting idea
  - One dataset for clustering, the other dataset for differential testing



TN test (Zhang et. al., Cell Systems 2019)

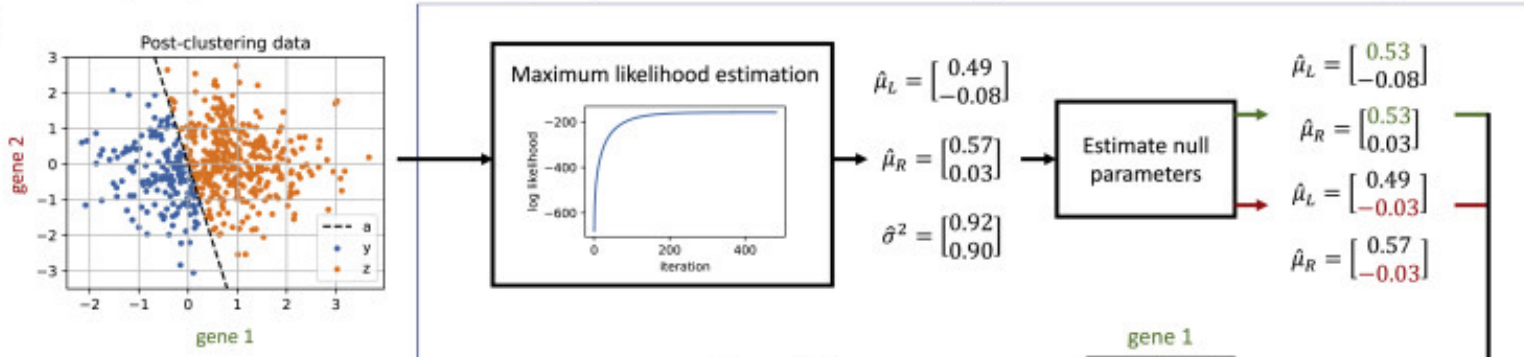
- Core steps
  - Clustering approximation on dataset 1
    - Apply any clustering algorithm to get the clustering result
    - When comparing between two clusters, use a linear hyperplane to approximate the clustering result

    - Benefit: the clustering result becomes a known truncation event on the test data

    - Apply the same clustering result on the dataset 2

TN test (Zhang et. al., Cell Systems 2019)

- Core steps
  - Clustering approximation on dataset 1
  - Truncated normal test on dataset 2
    - Fit truncated multivariate normal distribution on each cluster
    - Test for each gene g: H_{0g}: $\mu_{g1} = \mu_{g2}$
      - Estimate the null distribution (two-component Gaussian mixture) under each H_{0g}



  - Compute mean and variance of the truncated normals under the null distribution and compute the z-value to construct p-values

$$TN = \frac{m(\bar{z}_g - \mu_{Z_g}) - n(\bar{y}_g - \mu_{Y_g})}{\sqrt{m\sigma^2_{Z_g} + n\sigma^2_{Y_g}}} \xrightarrow{\text{CLT}} \mathcal{N}(0,1)$$

Data thinning (Neufeld et. al., Biostatistics 2024)

- A count splitting idea
- Key assumption

$$\mathbf{X}_{ij} \overset{\text{ind.}}{\sim} \text{Poisson}(\gamma_i \Lambda_{ij}), \qquad \log(\Lambda_{ij}) = \beta_{0j} + \beta_{1j} L_i, \qquad \beta_{1j}, L_i \in \mathbb{R},$$

  - X_{ij} observed scRNA-seq counts, L_i: unknown true cluster labels
  - This model is actually not enough as Λ_{ij} are not the true gene expressions (much less fluctuated and does not capture gene-gene dependence other than L_i)
- Key property

$$\mathbf{X}_{ij}^{\text{train}} \mid \{\mathbf{X}_{ij} = X_{ij}\} \overset{\text{ind.}}{\sim} \text{Binomial}(X_{ij}, \epsilon), \qquad X^{\text{test}} = X - X^{\text{train}}$$

Proposition 1 (Binomial thinning of Poisson processes (see Durrett 2019, Section 3.7.2)) If $\mathbf{X}_{ij} \sim Poisson(\gamma_i \Lambda_{ij})$, then $\mathbf{X}_{ij}^{\text{train}}$ and $\mathbf{X}_{ij}^{\text{test}}$, as constructed in Algorithm 1, are independent. Furthermore, $\mathbf{X}_{ij}^{\text{train}} \sim Poisson(\epsilon \gamma_i \Lambda_{ij})$ and $\mathbf{X}_{ij}^{\text{test}} \sim Poisson((1-\epsilon)\gamma_i \Lambda_{ij})$.

  - Given Λ_{ij}, training data and test data are independent
    - Get cluster labels of the cells from training data, test using test data
- Main drawback: the framework ignores extra gene-gene dependence not captured by L_i
- Main advantage: flexible to work for any “post-estimation” inference task

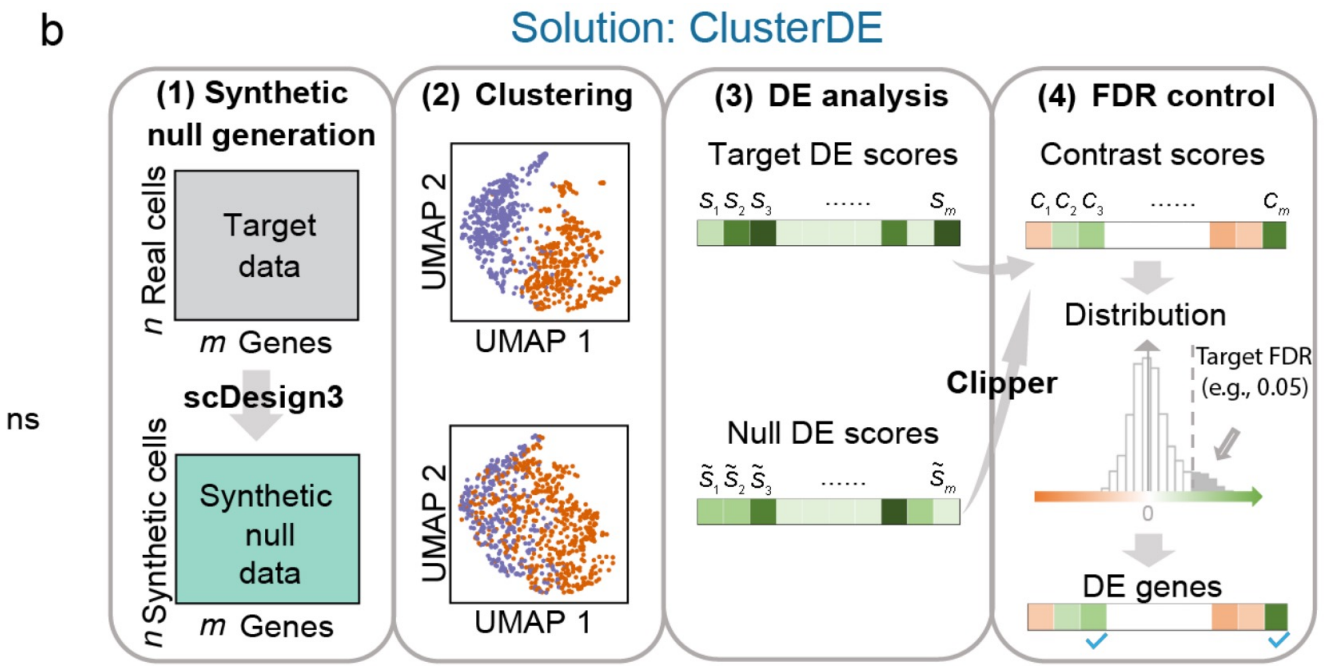
ClusterDE (Song et. al., BioRXiv 2024)

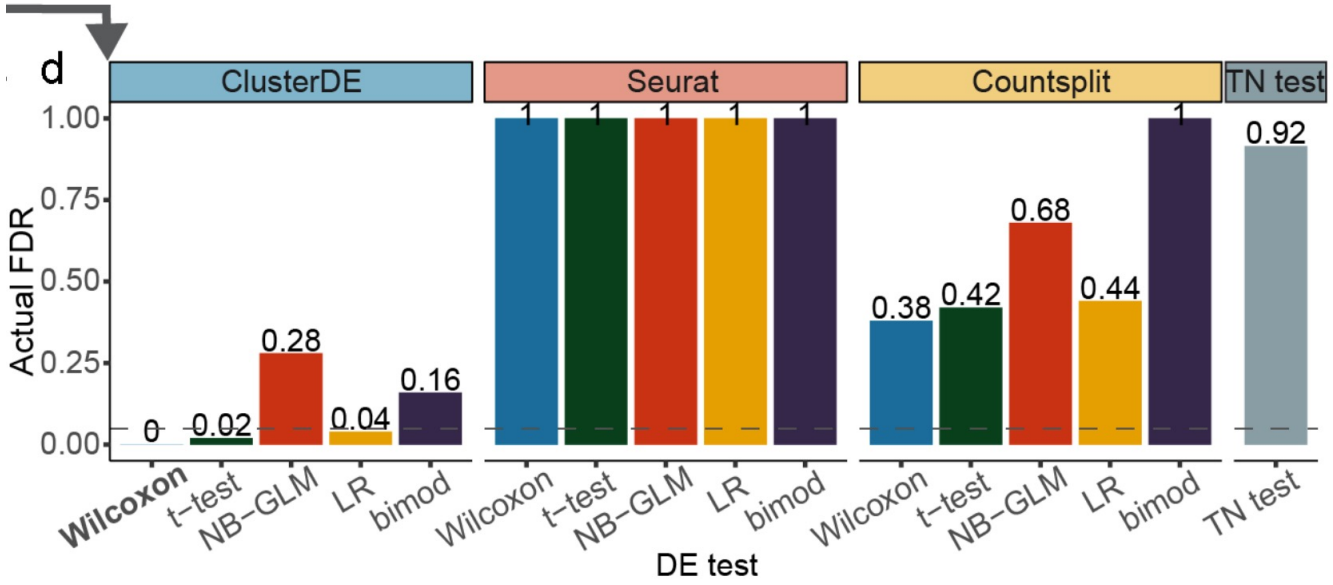
- Core idea:
  - Under the global null that the cell population is completely homogenous, generate synthetic data that match the real data distributions
    - Use scDesign3 to generate data:
      Synthetic data follows a Gaussian copula multivariate NB distribution, and matches mean, variance and gene-gene covariance with the real data
    - Use synthetic data to generate null distribution of test statistics for each gene
      - However, it is an invalid null distribution for the null $H_{0g}: \mu_{g1} = \mu_{g2}$ on real data

  - Apply clustering algorithm both on real data and synthetic data
    - As the synthetic data is generated under the global null, clustering algorithm results will be totally different from the real data
    - The method only work on two clusters at a time and allow the clustering algorithm to only generate two clusters

  - Calculate the same test statistics on real data and synthetic data to select differentially expressed genes
    - Instead of calculating the null distribution by generating multiple synthetic dataset, used a symmetric idea (similar to knockoff) for multiple test using only one synthetic data

ClusterDE (Song et. al., BioRXiv 2024)

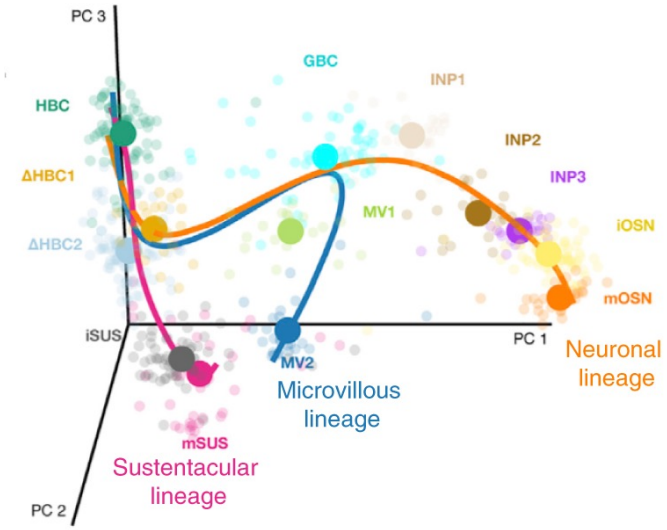


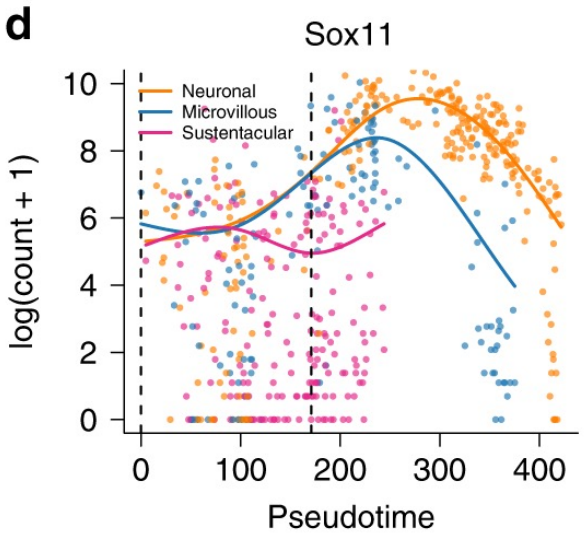


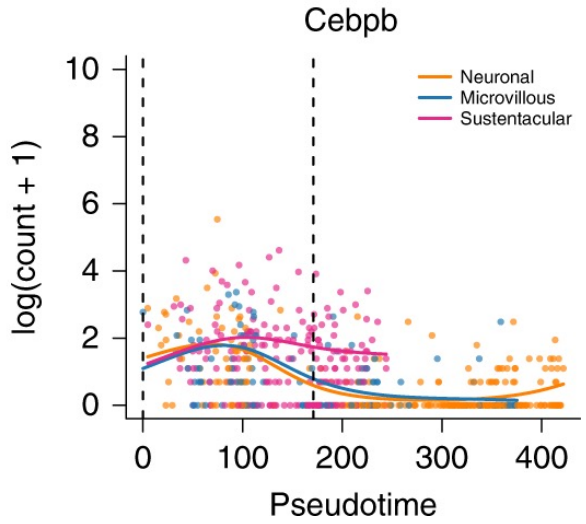
Results from homogenous cell population data simulated by scDesign3

Post trajectory inference differential testing

- After trajectory inference, researchers can be interested in different testing tasks:
  - Gene expression change along the pseudotime (for a specific lineage or sub-trajectory)
  - Differential gene expression between two lineages







Harder tasks:

- Whether an estimated branching event is true or false
- Whether the trajectory structure is different under two different conditions

tradeSeq (Berge et. al. 2020)

- For a specific gene g, using a generalized additive model (GAM) to describe how the observed count Y_{gi} for cell i depends on the pseudotime, lineage and other covariates U_i

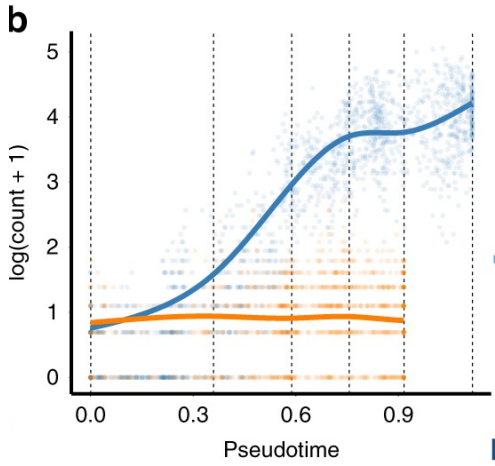
$$\begin{cases} Y_{gi} \sim NB(\mu_{gi}, \phi_g) \\ \log(\mu_{gi}) = \eta_{gi} \\ \eta_{gi} = \sum_{l=1}^{L} s_{gl}(T_{li}) Z_{li} + \mathbf{U}_i \boldsymbol{\alpha}_g + \log(N_i) \end{cases}$$

  - T_{li}: pseudotime of cell i, may depend on the lineage l
  - Z_{li}: binary lineage indicator of the cell
  - N_i: library size

  - $s_{gl}(t)$: natural cubic spline function (basis functions shared across all genes and lineages)

$$s_{gl}(t) = \sum_{k=1}^{K} b_k(t) \beta_{glk}$$

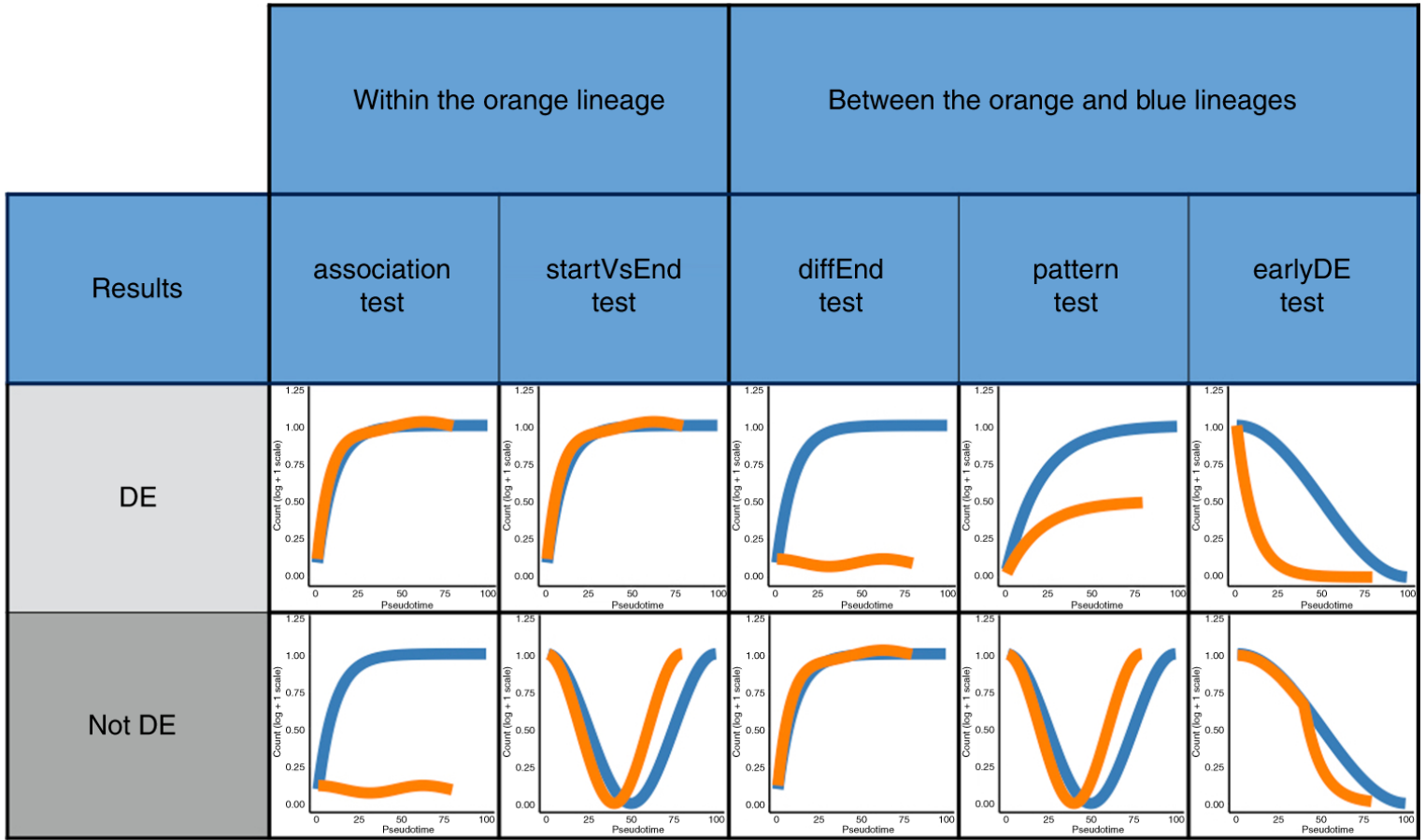
    - K selected by AIC (default K = 6 correspond to 6 knots)
    - Knots placed at even quantiles of the estimates pseudotime



tradeSeq (Berge et. al. 2020)

Test for differentially expressed genes

- Test if a gene change along the pseudotime $H_0: \beta_{glk} = \beta_{glk'}$ for any k, k'
- Test if a gene change between lineages: test if the mean gene expression change in any of the pseudotime from a set of possible scaled pseudotimes
- Compute p-values based on the wald statistics



Post estimating bias in testing after TI

- tradeSeq treat the estimated pseudotime T_i and and lineage positioning Z_{li} as known
- This can create a double-dipping issue

- Idea null hypothesis: $H_{0g}: T_i \perp Y_{ig}$
  - Challenge: T_i is not observed, we can only obtain an estimate $\hat{T}_i = \hat{f}(Y_{i.})$ by TI
    - Much more false positives compared to clustering as pseudotime estimation (estimate an ordering of the cells) is always much noisier

  - We would like to account for the uncertainty in $\hat{T}_i$
    - Unsupervised learning: T_i is never observed, if $\hat{T}_i$ is terribly estimated, then we will never be able to test H_{0g}

      - A clear statement of a reasonable H_{0g} or requirement of nice property of $\hat{T}_i$ seems necessary

data thinning (Neufeld et. al., Biostatistics 2024)

$$\mathbf{X}_{ij}^{\text{train}} \mid \{\mathbf{X}_{ij} = X_{ij}\} \overset{\text{ind.}}{\sim} \text{Binomial}\,(X_{ij}, \epsilon), \quad X^{\text{test}} = X - X^{\text{train}}$$

- Perform trajectory inference and estimate pseudotime on the training data and perform differential testing on the test data
- Assume the model

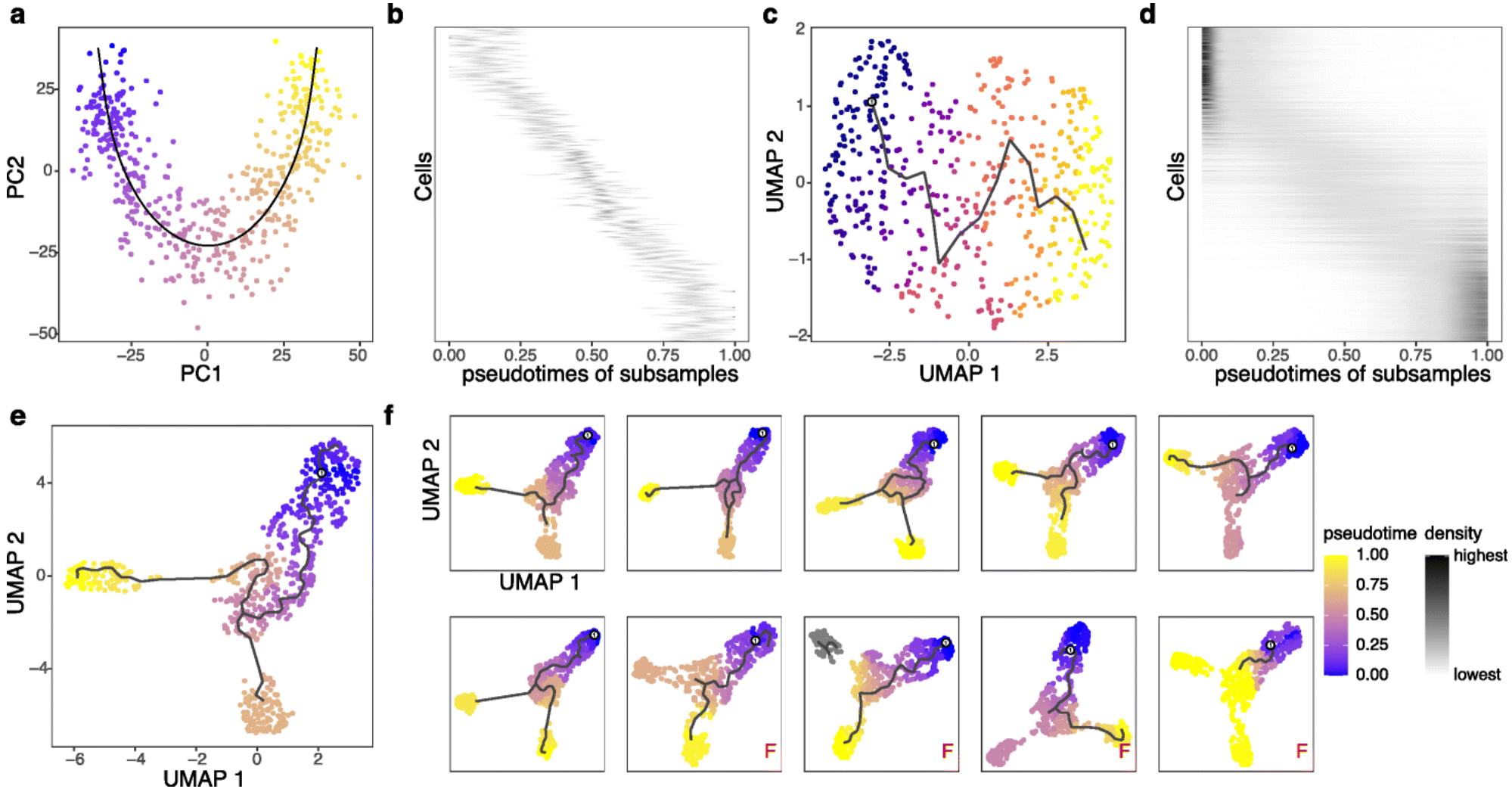
$$\mathbf{X}_{ij} \overset{\text{ind.}}{\sim} \text{Poisson}(\gamma_i \Lambda_{ij}), \qquad \log(\Lambda_{ij}) = \beta_{0j} + \beta_{1j} L_i, \qquad \beta_{1j}, L_i \in \mathbb{R},$$

- Pros:
  - allow any trajectory inference methods
  - Computationally cost effective

- Cons:
  - Assume that gene-gene dependence are completely captured by the pseudotime
  - Estimated trajectory structure and cell ordering can be very different if reducing the sequencing depth by a half

PseudotimeDE (Song and Li, Genome Biology 2021)

- Idea: subsampling can evaluate the variation of the estimated pseudotime



PseudotimeDE (Song and Li, Genome Biology 2021)

Core steps:

- Sub-sample 80% of the cells each time to create multiple versions of the “data”
- Apply trajectory inference method both on the real data and on each subsampled data

- To test $H_{0g}: T_i \perp Y_{ig}$, the method creates the null data by permuting the estimated pseudotime on sub-sampled data

- Then the same GAM model is fitted on each gene and permuted pseudotime to create a null distribution of the test statistics of H_{0g}

- The real test statistics is compared with the null distribution to compute a p-value

- Main con: the permuted pseudotime does not have dependence on gene g, while on real data there is such dependence, thus the null distribution does not reflect the double dipping bias

- Evaluation of the performance is hard as it is challenging to create data with known trajectory structure, known DE genes and realistic gene-gene dependence
  - The empirical performance of the method is surprisingly not bad on simulated data

Statistical inference and estimation after denoising

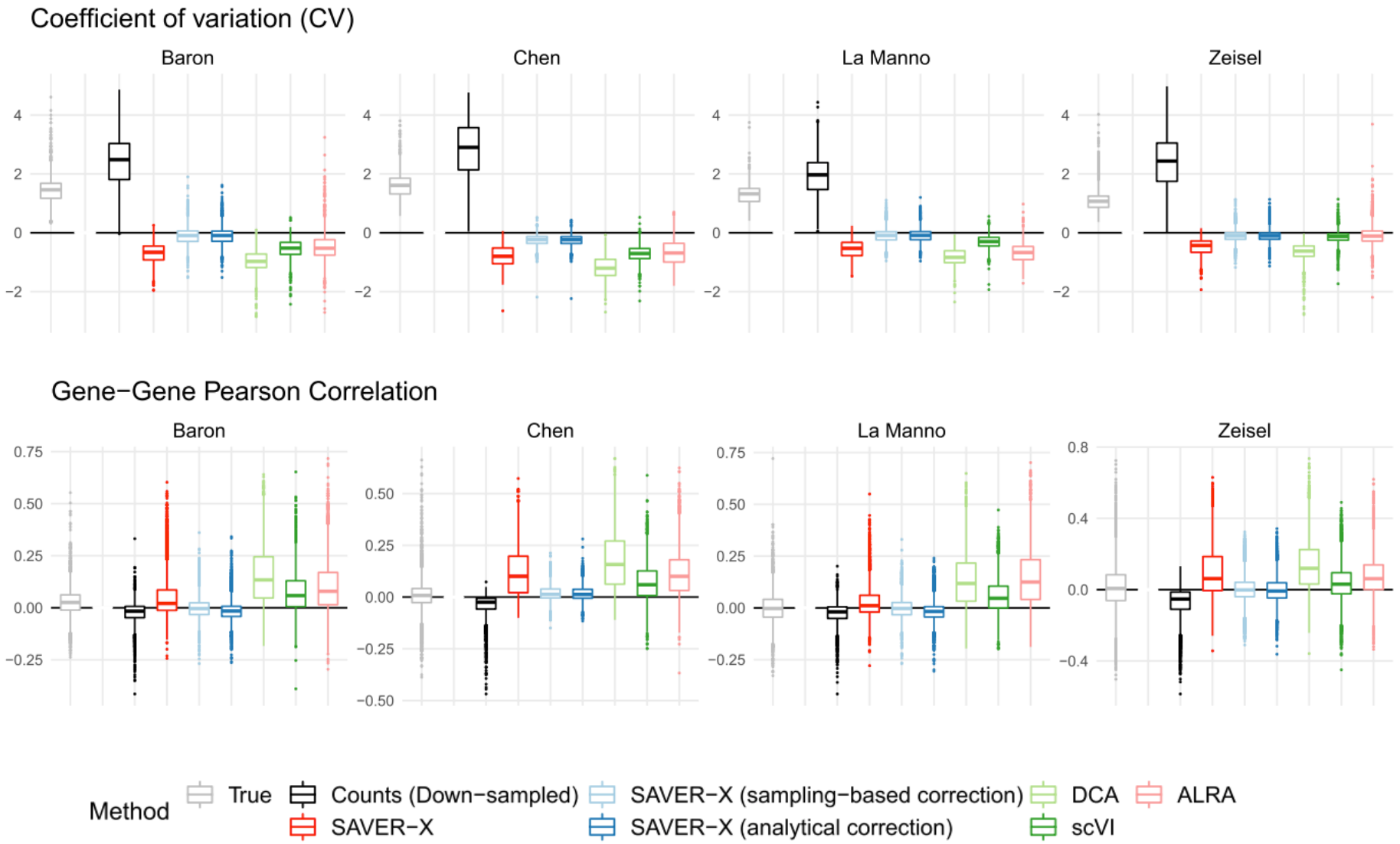
Estimation and inference after scRNA-seq denoising:

- Ideally, denoising provides estimation of the underlying true gene expression

- However, the denoised data
  - Introduce dependence between cells which are originally independently sampled
    - Standard differential testing between two cell types can introduce false positives because of cell-cell dependence

  - May be over-smoothed so that the variability across cells are less than the true gene expression variability and the gene-gene dependence may be higher
    - Can lead to biased estimation in gene properties

Bias in estimating gene properties



- DCA, ALRA, scVI: autoencoder output or SVD (with thresholding)
- SAVER-X: weighted average between autoencoder output and observed data to compute posterior mean

Correcting for bias in estimating gene properties

(Agarwal et. al. Statistical Science 2020)

- Hierarchical model

$$Y_{gc}|X_{gc} \sim \text{Poisson}(\alpha_{gc}X_{gc}) \qquad X_{gc}|\Lambda_{gc} \overset{\text{indep}}{\sim} F(\Lambda_{gc}, \varphi_g\Lambda_{gc})$$

  - Λ_{gc}: structured part of the true gene expression (low rank, autoencoder output …)

  - $f(X)$: gene property of interest, mean, variance, gene-gene correlation …
  - Goal: estimate $E[f(X) \mid Y, \Lambda]$.

- General solution for any $f(X)$:
  - denoising method like SAVER or SAVER-X estimates posterior distribution (gamma distribution) of X
  - Repeatly sample from the posterior distribution, calculate $f(X)$ and compute the mean

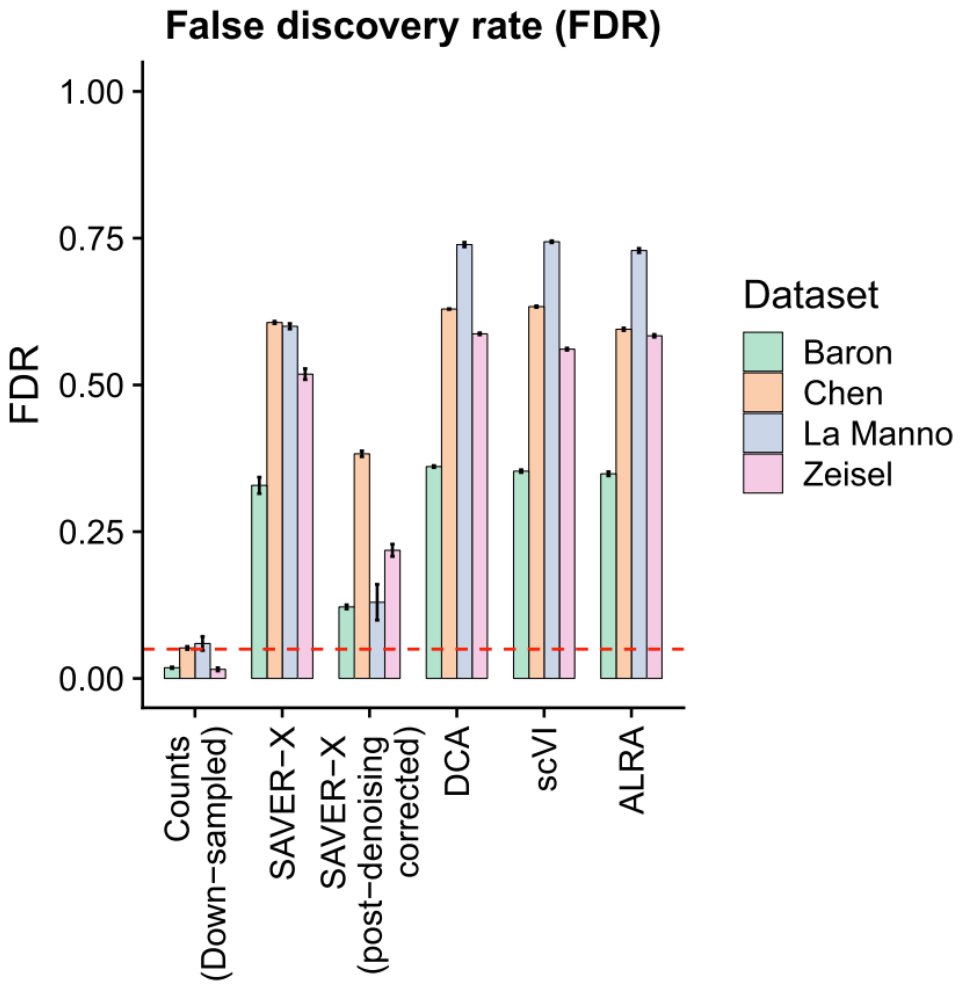
- Analytical solution for special $f(X)$:
  - Variance of a single gene $E[V_g(X) \mid Y, \Lambda]$

$$\approx \frac{1}{C}\left[\sum_{c=1}^{C}(\widehat{X}_{gc} - \bar{\widehat{X}}_{g\cdot})^2 + \sum_{c=1}^{C}\widehat{v}_{gc}\right]$$

False positives in differential gene testing

- Severe problem first discussed in Andrews and Hemberg, F1000Research 2018
- Finding a solution is really challenging

- Previous posterior justification won't work as the posterior is given Λ_{gc} which is estimated from the data and can introduce cell-cell dependence



Related papers

- Gao, L. L., Bien, J., & Witten, D. (2024). Selective inference for hierarchical clustering. Journal of the American Statistical Association, 119(545), 332-342.
- Chen, Y. T., & Witten, D. M. (2023). Selective inference for k-means clustering. *Journal of Machine Learning Research*, *24*(152), 1-41.
- Zhang, J. M., Kamath, G. M., & David, N. T. (2019). Valid post-clustering differential analysis for single-cell RNA-Seq. Cell systems, 9(4), 383-392.
- Neufeld, A., Gao, L. L., Popp, J., Battle, A., & Witten, D. (2024). Inference after latent variable estimation for single-cell RNA sequencing data. Biostatistics, 25(1), 270-287.
- Song, D., Li, K., Ge, X., & Li, J. J. (2023). ClusterDE: a post-clustering differential expression (DE) method robust to false-positive inflation caused by double dipping. Research Square.

- Van den Berge, K., Roux de Bézieux, H., Street, K., Saelens, W., Cannoodt, R., Saeys, Y., ... & Clement, L. (2020). Trajectory-based differential expression analysis for single-cell sequencing data. Nature communications, 11(1), 1201.
- Song, D., & Li, J. J. (2021). PseudotimeDE: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell RNA sequencing data. Genome biology, 22(1), 124.
- Agarwal, D., Wang, J., & Zhang, N. R. (2020). Data denoising and post-denoising corrections in single cell RNA sequencing.
- Andrews, T. S., & Hemberg, M. (2018). False signals induced by single-cell imputation. *F1000Research*, *7*.