Clustering and normalization of single-cell RNA-seq data with BISCUIT

(Bayesian Inference for Single-cell ClUstering and ImpuTing)

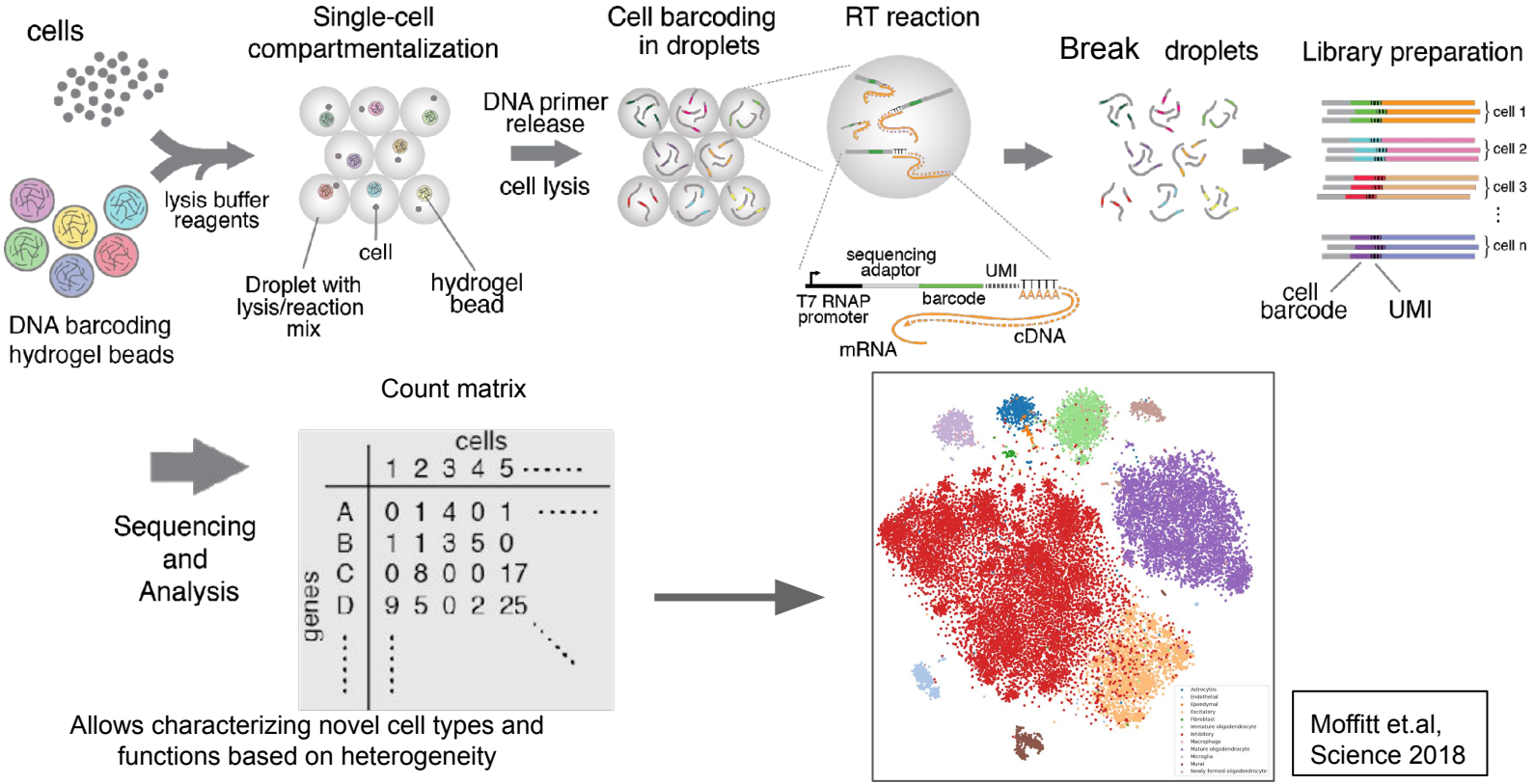
Sandhya Prabhakaran

Integrated Mathematical Oncology Department
Moffitt Cancer Center, Florida

[This presentation deals with work done while in the lab of Dr. Dana Pe'er]

Single-cell RNA-seq reveals heterogeneity in expression

Measurement of gene expression at resolution of single cells

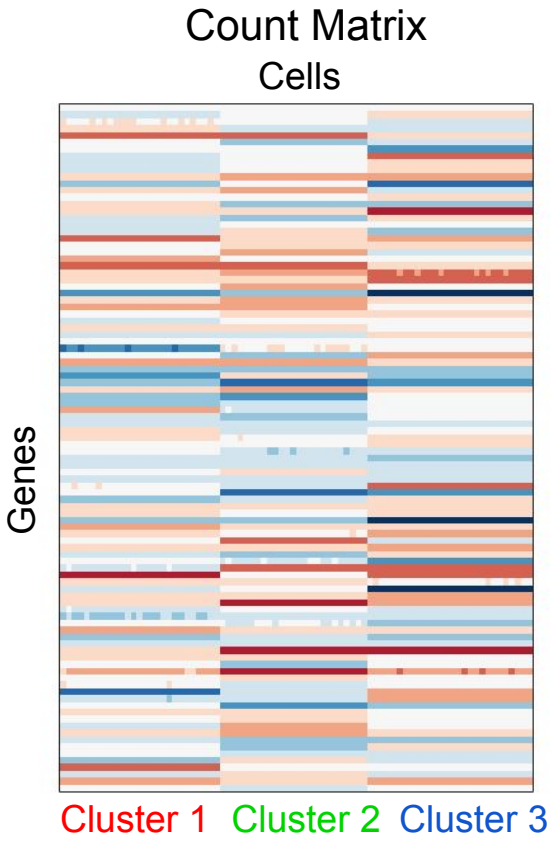


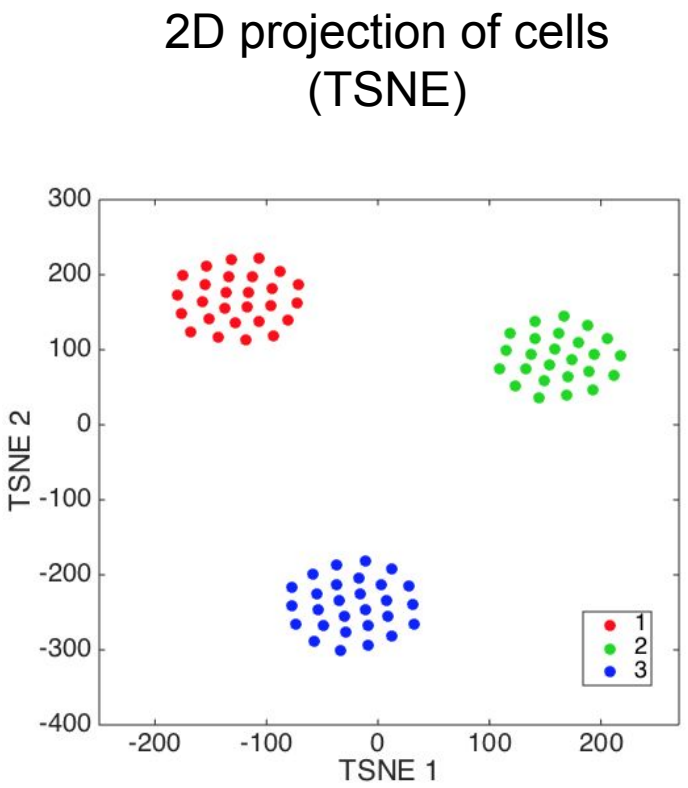
(indrop (Klein et al *Cell* 2015))

Problems with Single-cell RNA-seq data

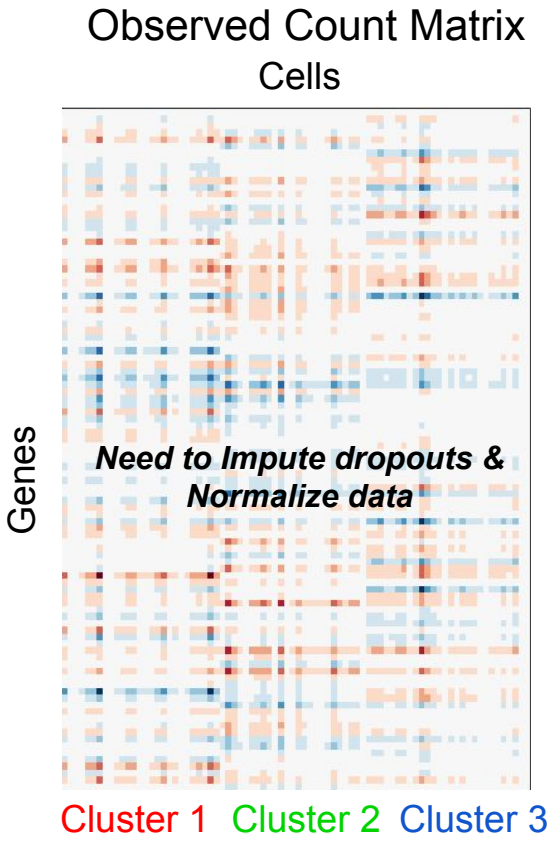
- Sampling sparse amounts of mRNA leads to “Drop-outs”
- Amplification differences
- Cell-type specific capture rates
- Significant technical variation - mistaken for biological heterogeneity

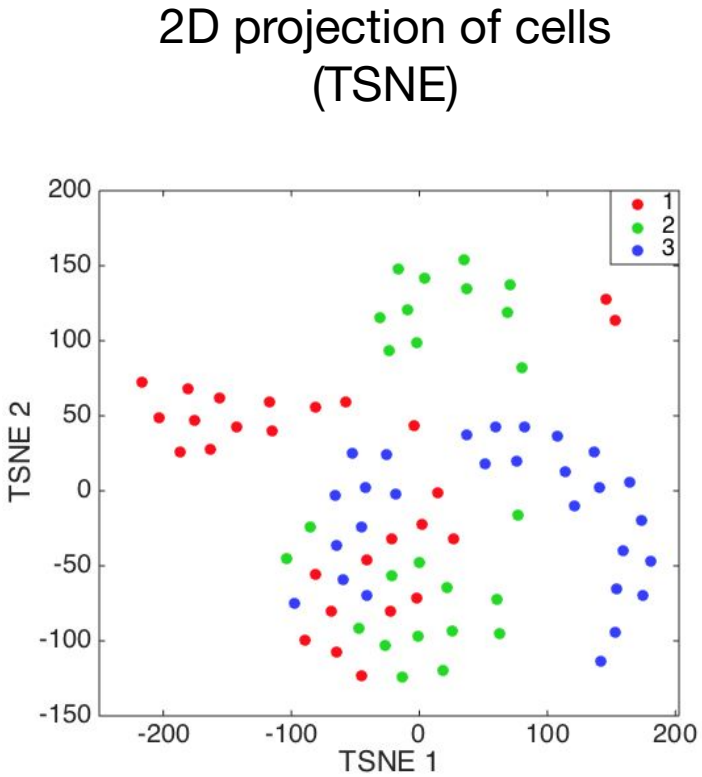
Goal: Characterizing cell subpopulations using Single-cell RNA-seq data



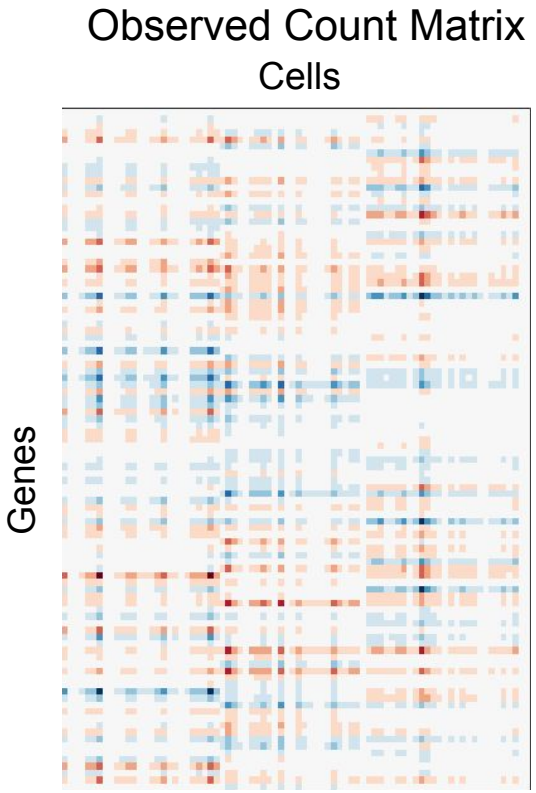


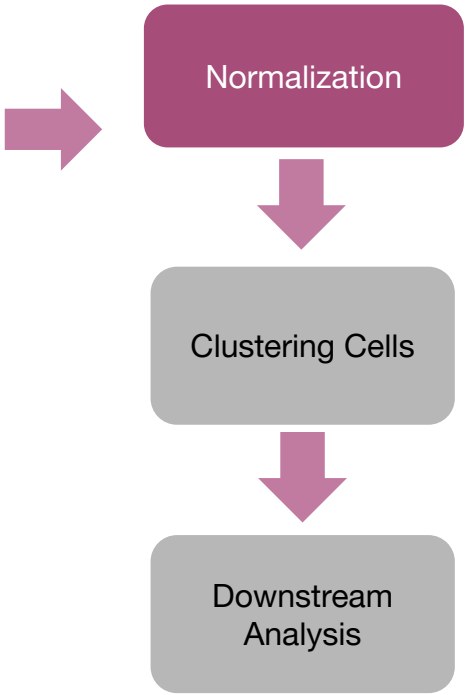
Problems: Single-cell RNA-seq data involves significant dropouts and library size variation





Common Approach: Normalizing independent of cell types





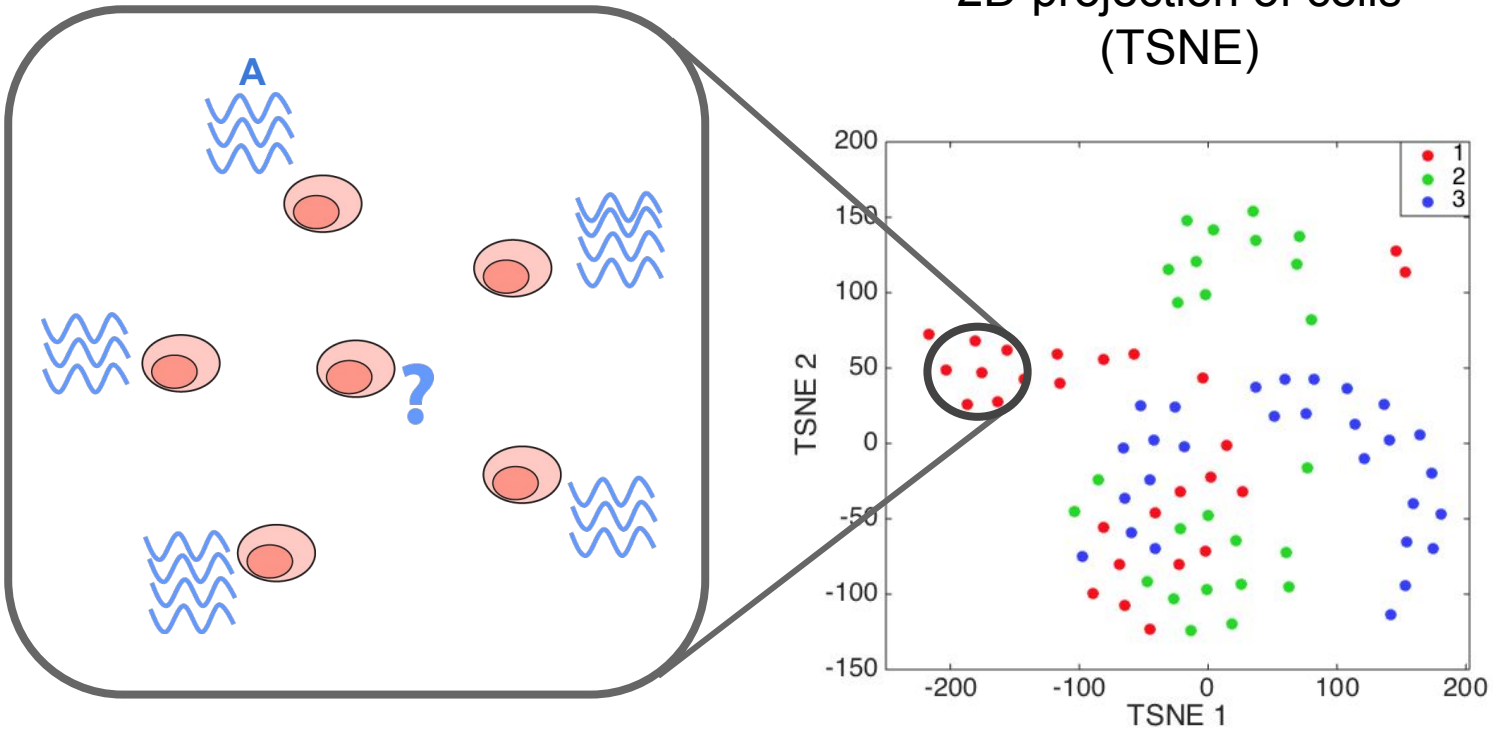
To mean/median library size
Downsampling
BASiCS with spike-ins/ERCCs

Problems:

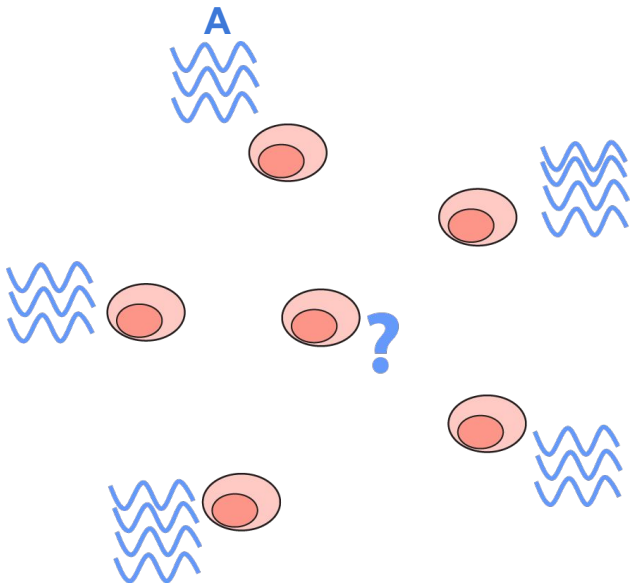
- **Dropouts not resolved**
  Zeros remain zero!
- Removes biological stochasticity specific to cell type
- Leads to improper clustering; Biased downstream analysis

Main Concepts behind BISCUIT for Normalization and Imputing

Two ideas for imputing expression in Single-cell RNA-seq data



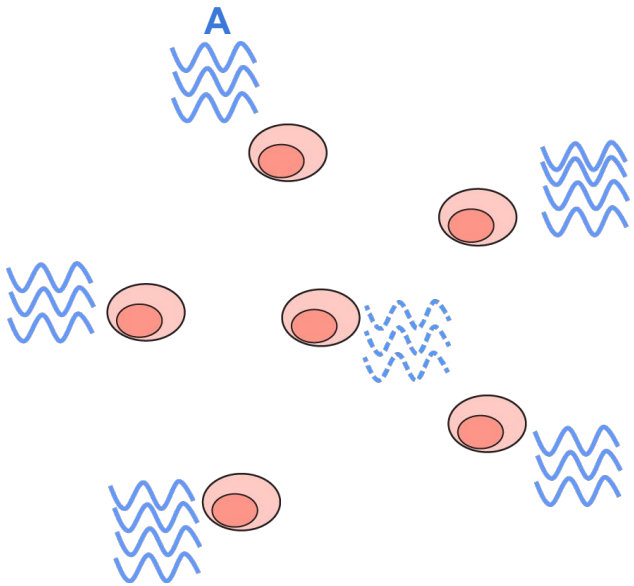
Idea 1: Impute dropouts based on cell type



No expression of **Gene A** in a cell

But we observe cells with same type mostly have high expression of Gene A

Idea 1: Impute dropouts based on cell type

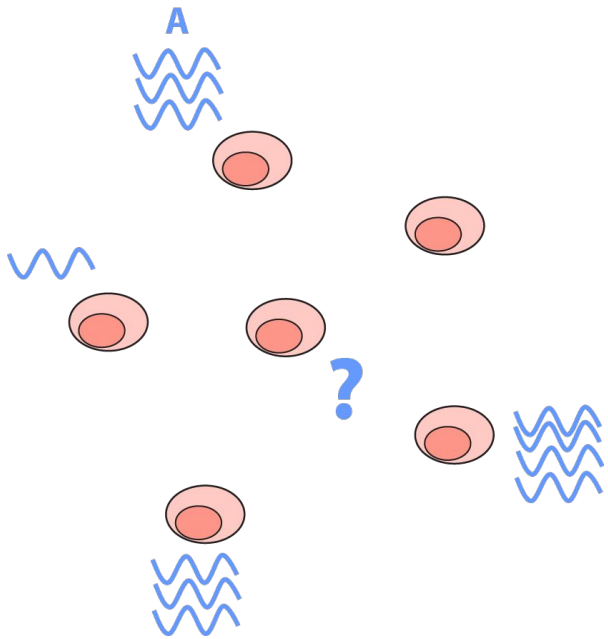


No expression of **Gene A** in a cell

But we observe cells with same type mostly have high expression of Gene A

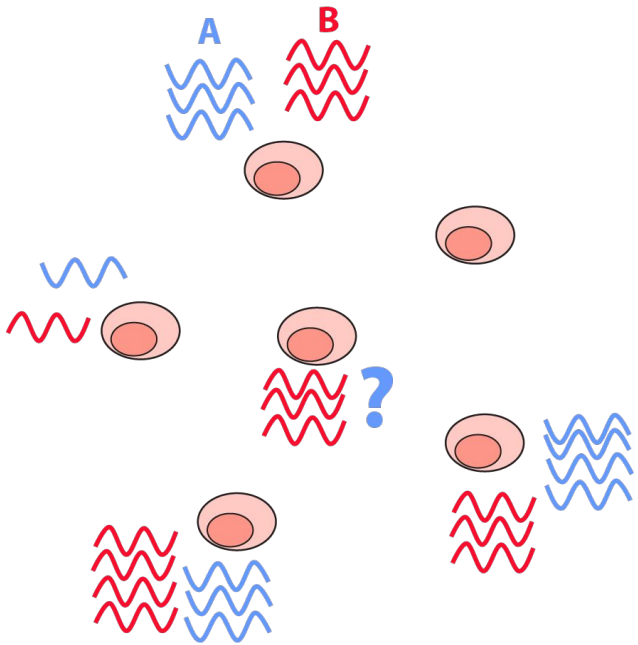
Impute dropout in Gene A based on similar cells

Idea 2: Impute dropouts based on co-expression patterns



No significant inference based on similar cells

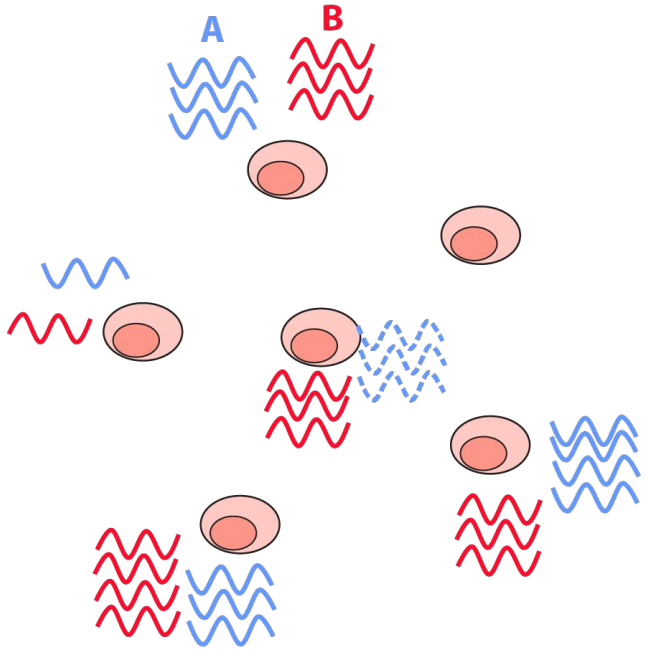
Idea 2: Impute dropouts based on co-expression patterns



No significant inference based on similar cells

However **Gene A** always co-expressed with Gene B in cells of same type

Idea 2: Impute dropouts based on co-expression patterns

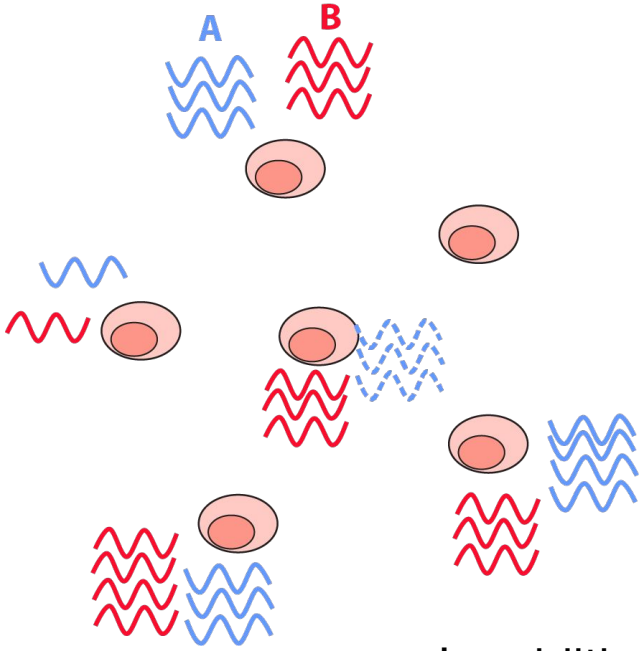


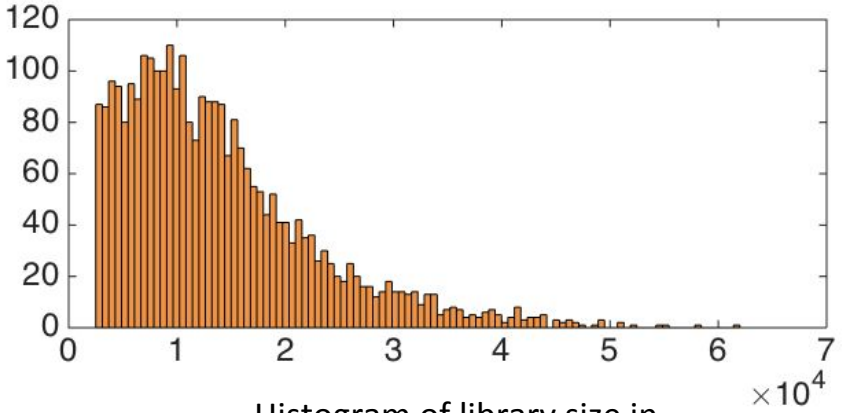
No significant inference based on similar cells

However **Gene A** always co-expressed with Gene B in cells of same type

Impute dropout in Gene A based on Gene B

Normalization of Single-cell RNA-seq data

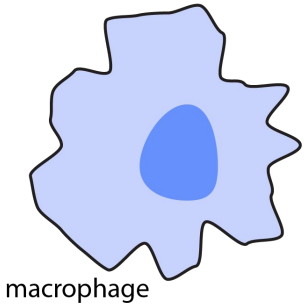


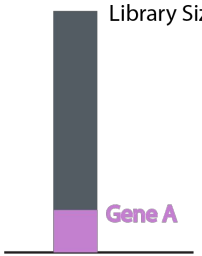


Histogram of library size in example SC dataset
From Zeisel, Science 2014

In addition to imputing dropouts,
we need to **normalize** data by library size

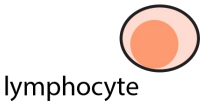
Problem with Global Normalization

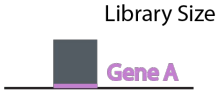




Example Housekeeping Gene

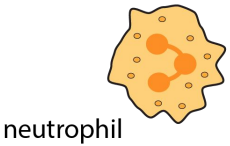
Cells with different sizes have very different total number of transcripts

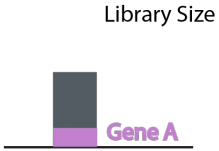




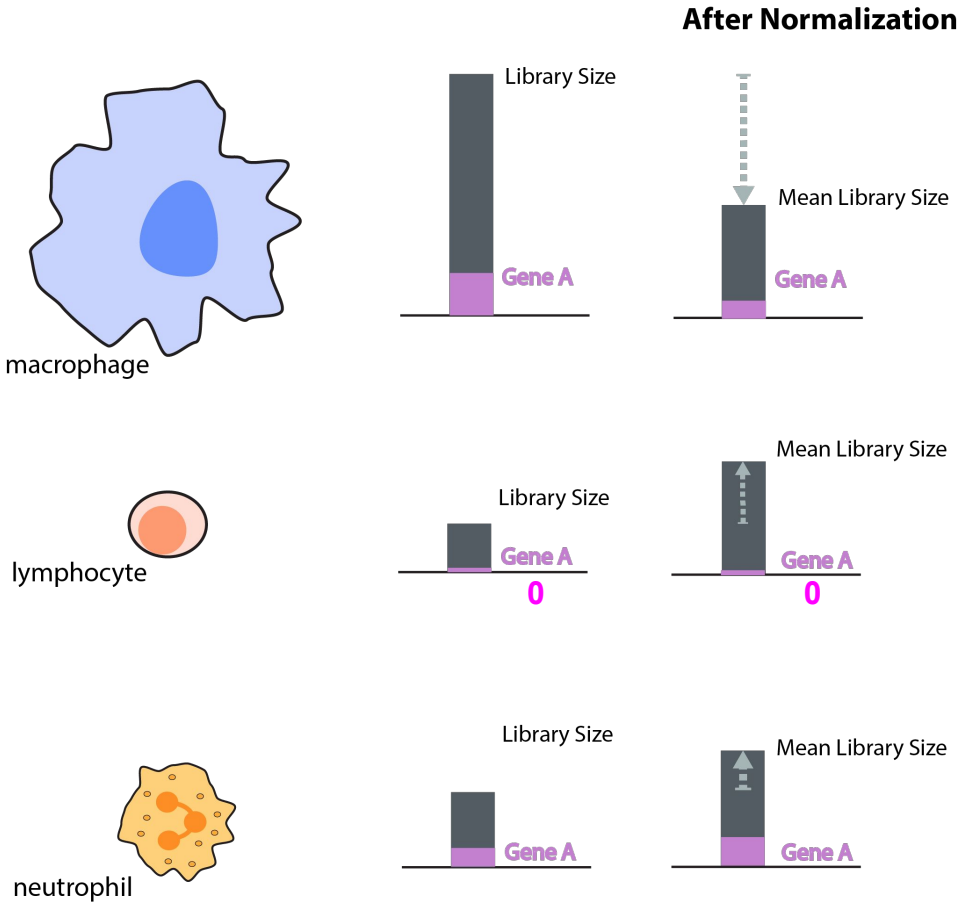
0

High chance of Dropouts in smaller cells





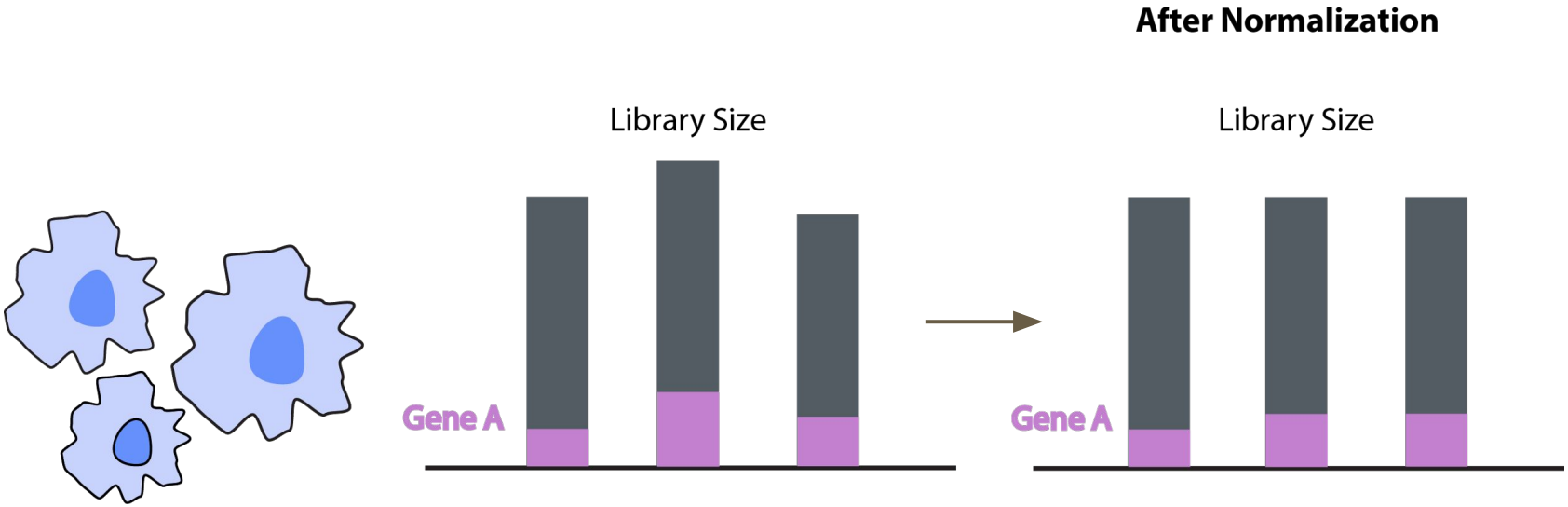
Problem with Global Normalization



Dropout not resolved

Spurious Differential Expression

Key: Different normalization for each cell type



Chicken and egg problem:
Normalize based on cell types but we do not know cell types!

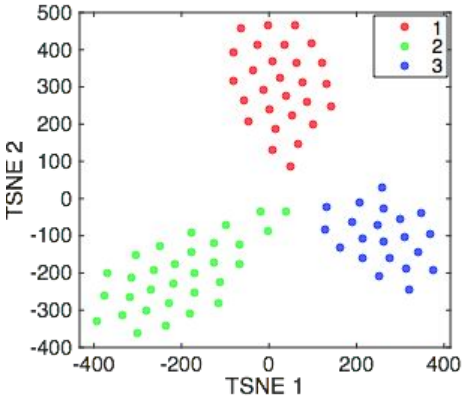
Approach:

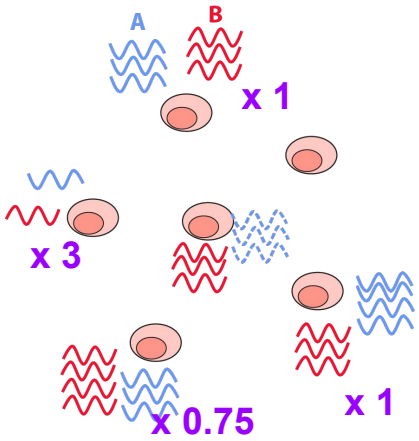
Simultaneous inference of clusters and imputing parameters

Clustering Cells

iterative learning

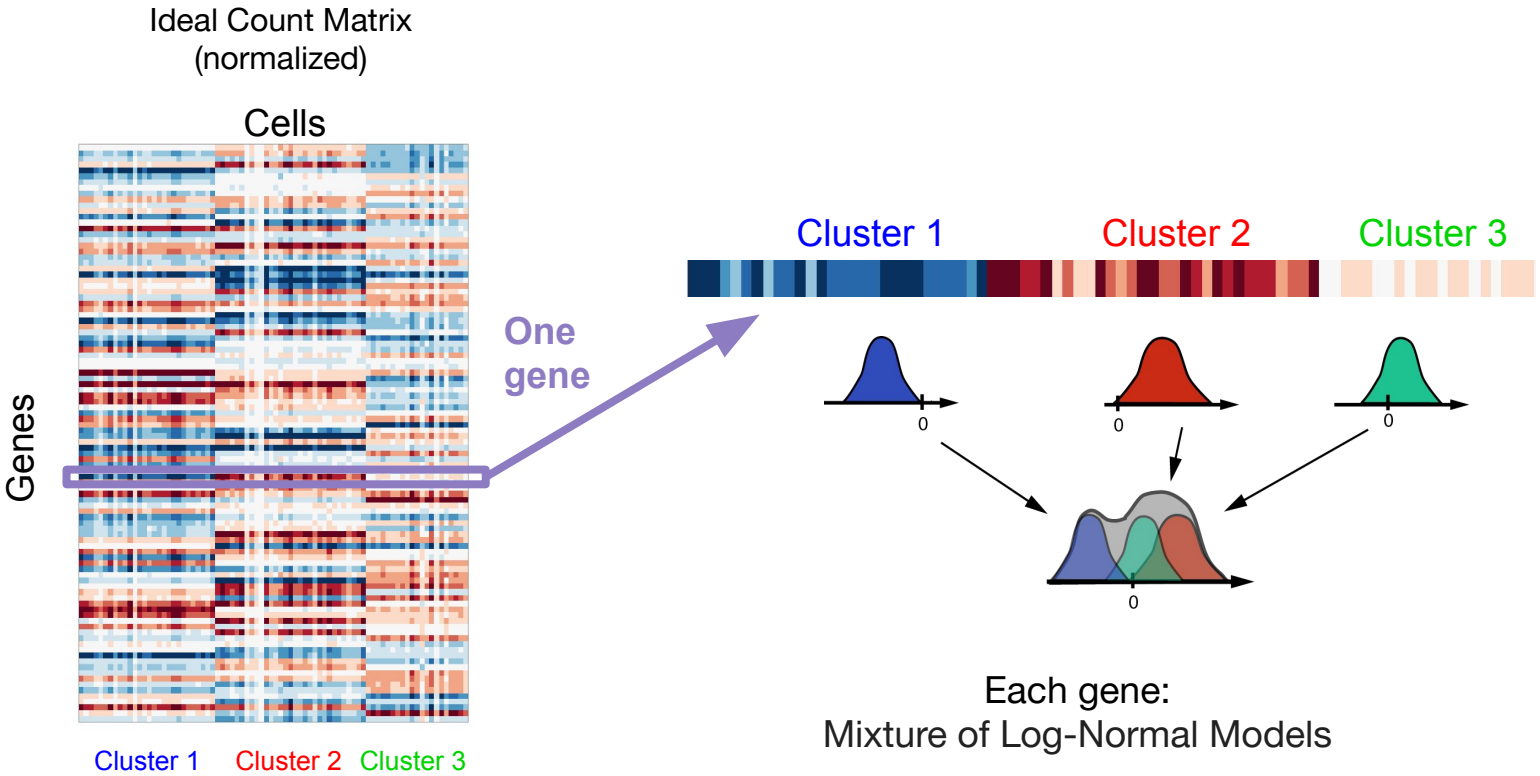
Imputing & Normalization



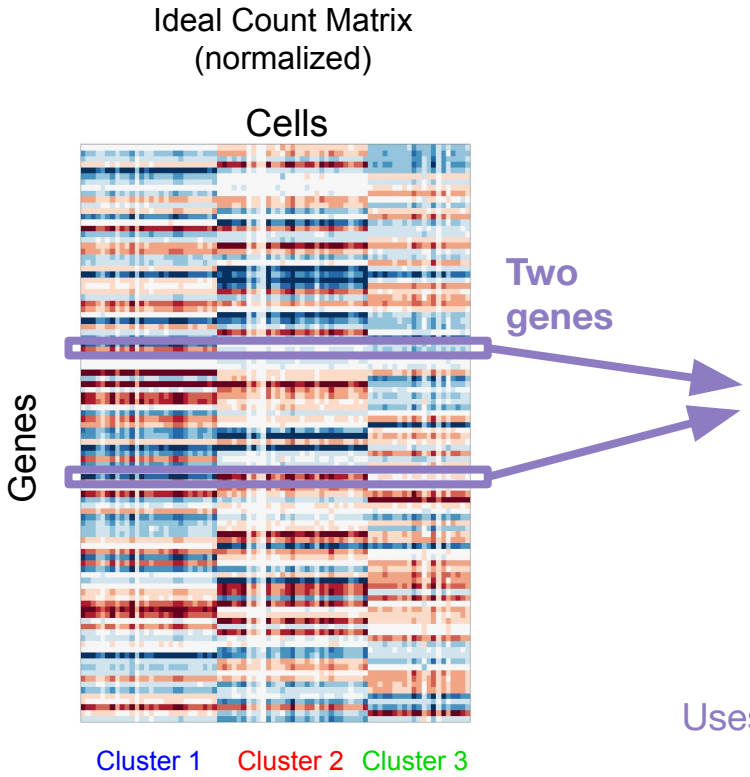


Modeling Single-cell data using a Hierarchical Bayesian Mixture Model

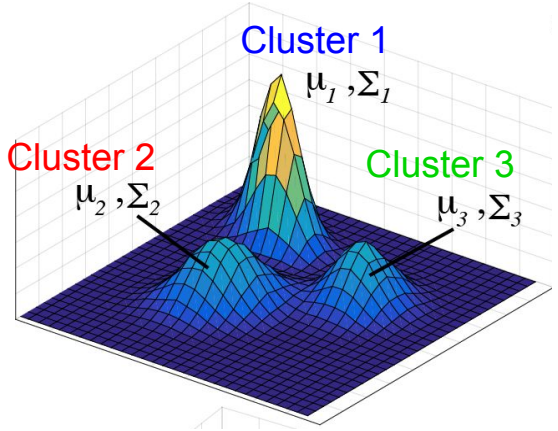
Modeling Clusters of Cells using a Bayesian Mixture Model



Modeling Clusters of Cells using a Bayesian Mixture Model

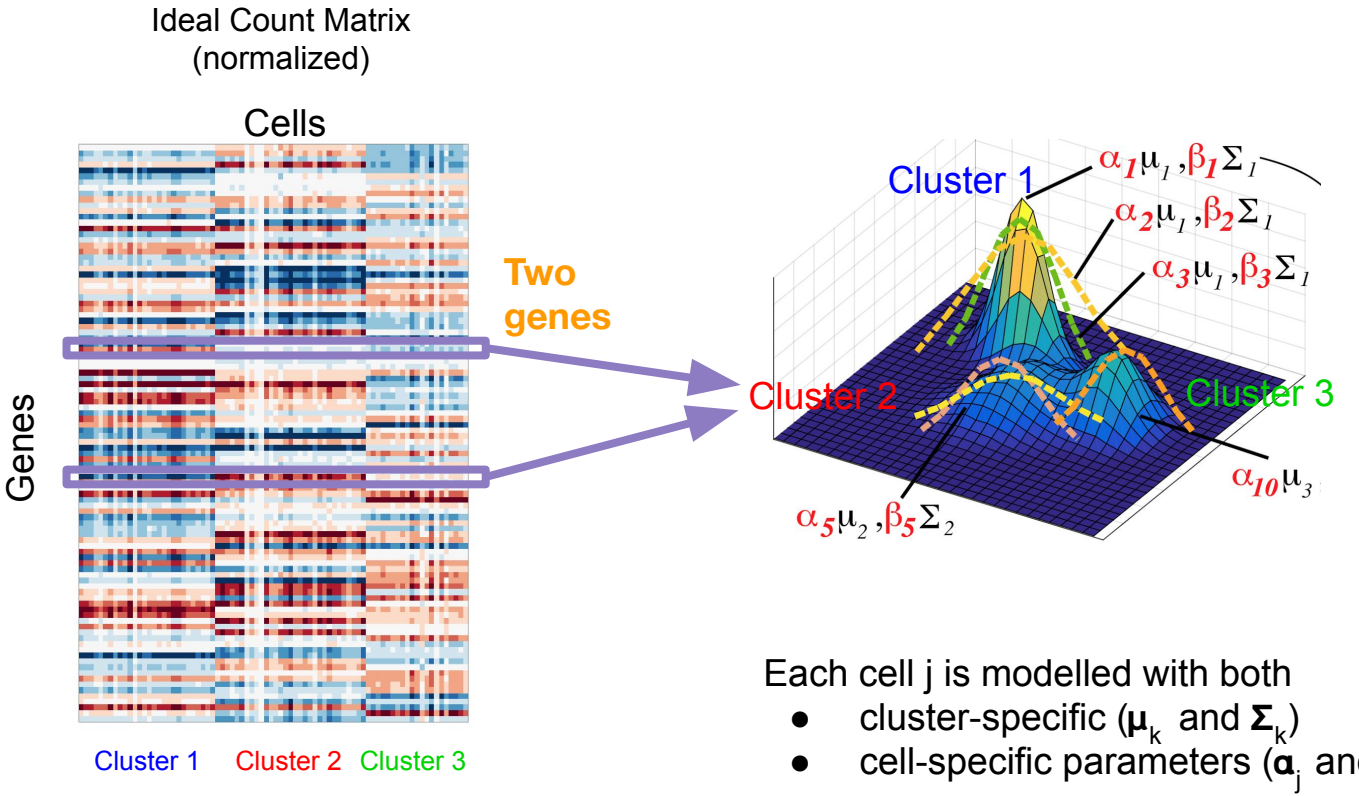


$$\boldsymbol{y}_j \sim \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k) \qquad z_j = k$$



Modeling all genes together:
Mixture of Multivariate Log-Normals
Uses gene co-expression patterns for clustering and imputing

Modeling Clusters of Cells using a Bayesian Mixture Model

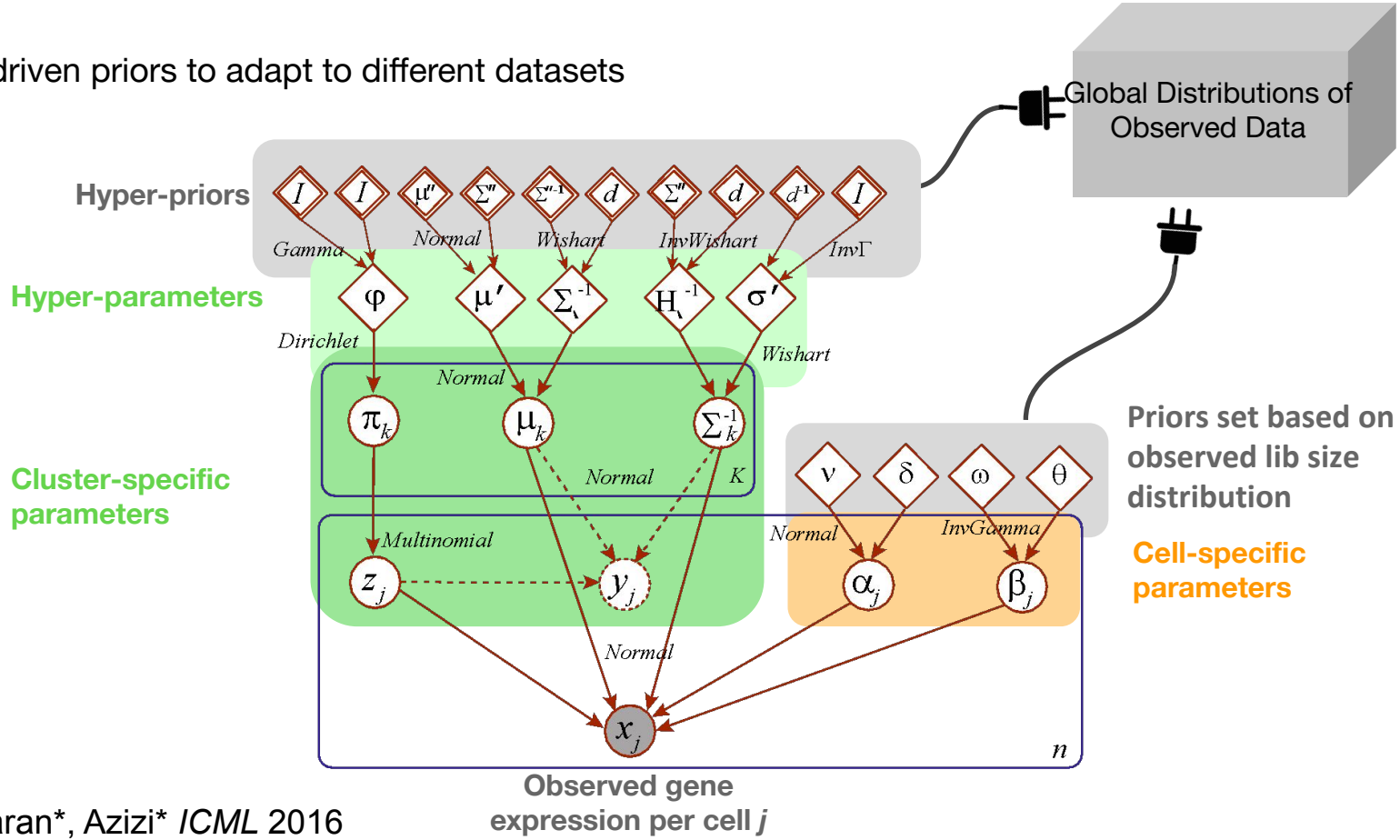


Each cell j is modelled with both

- cluster-specific ($\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$)
- cell-specific parameters ($\boldsymbol{\alpha}_j$ and $\boldsymbol{\beta}_j$)

BISCUIT (Bayesian Inference for Single-cell ClUstering and ImpuTing)

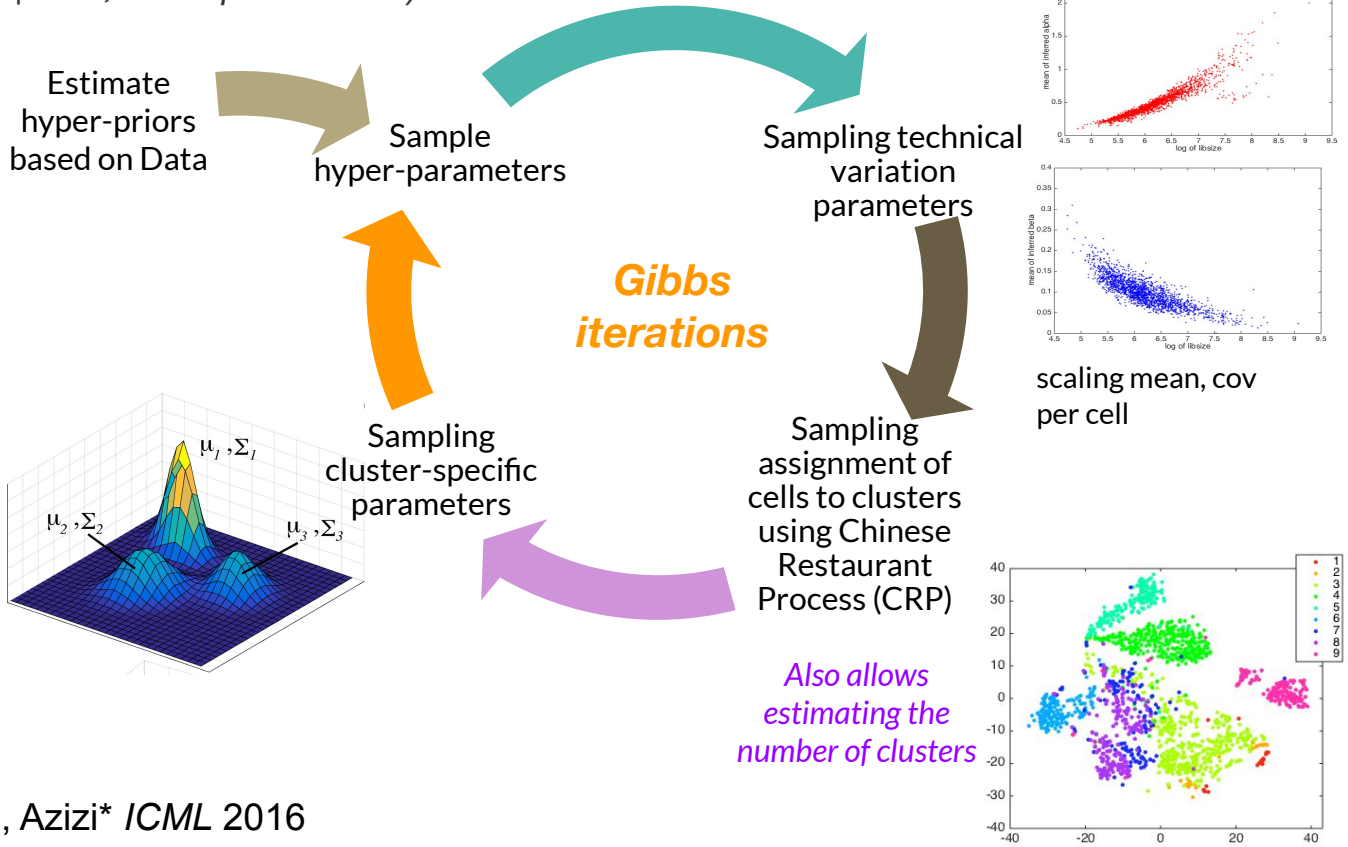
Data-driven priors to adapt to different datasets



Prabhakaran*, Azizi* *ICML* 2016

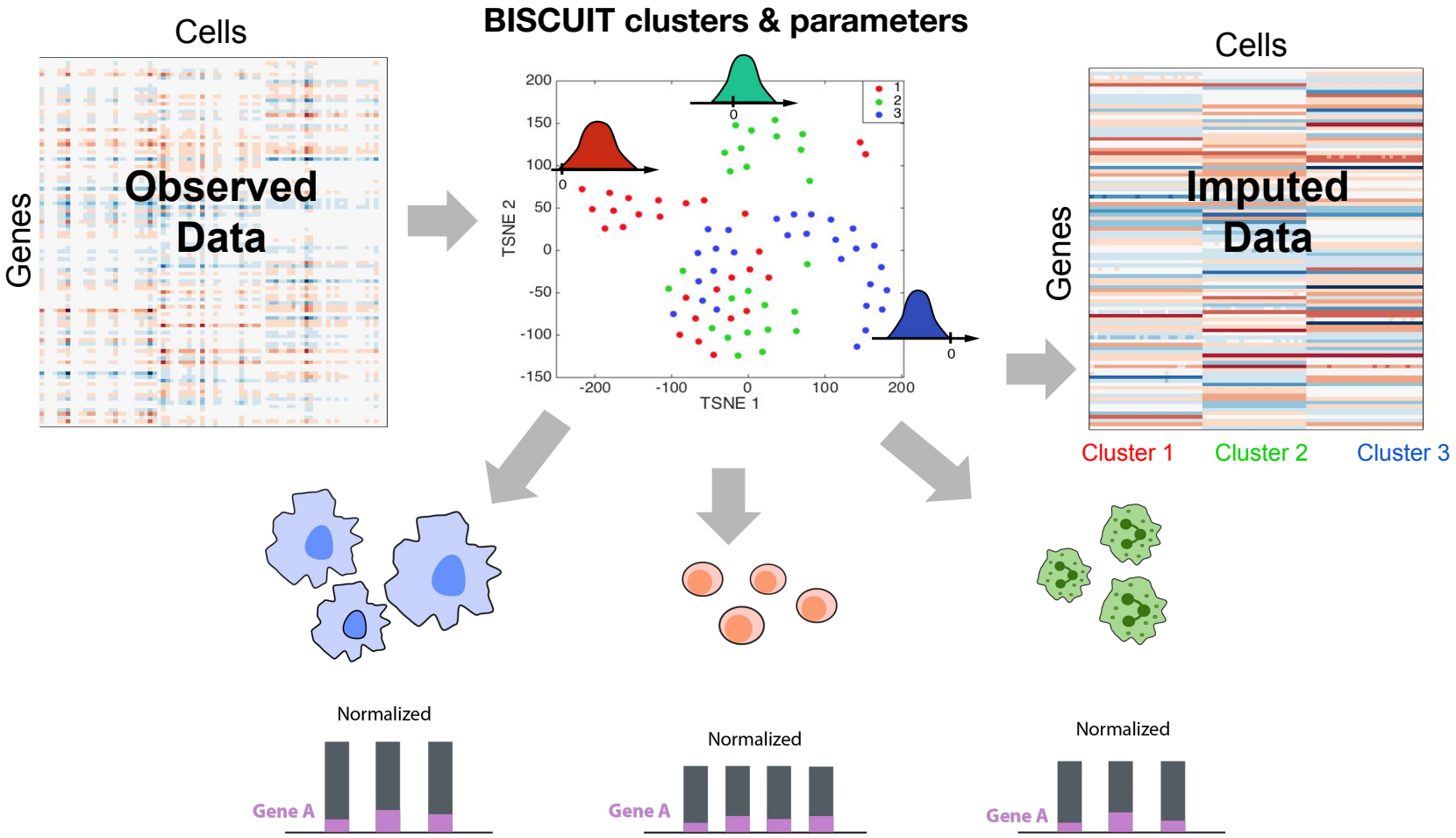
Inference Algorithm

Parallel Sampling from derived conditional posterior distributions:
P(parameter| data, other parameters)



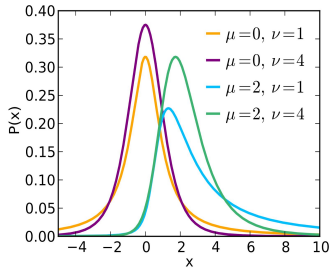
Prabhakaran*, Azizi* *ICML* 2016

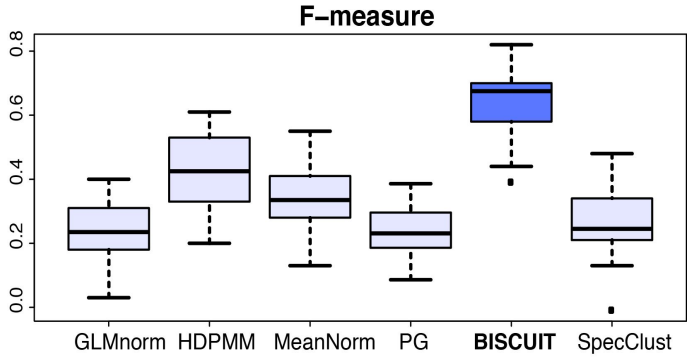
Cluster-dependent Imputing & Normalizing



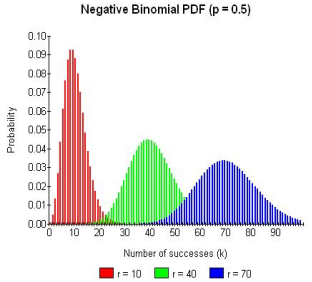
Model Mismatch:
Robustness when counts are not LogNormal

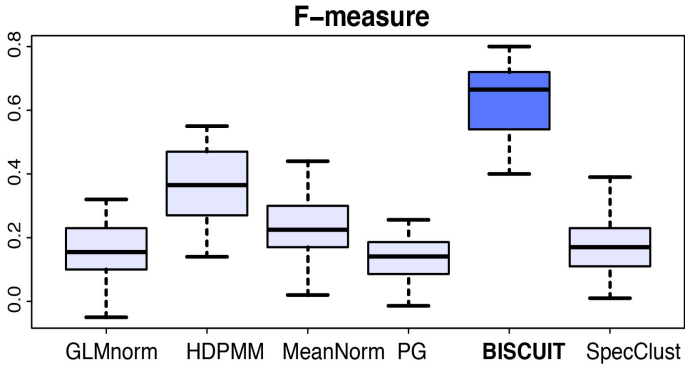
Noncentral Student's t





(Log) Negative binomial





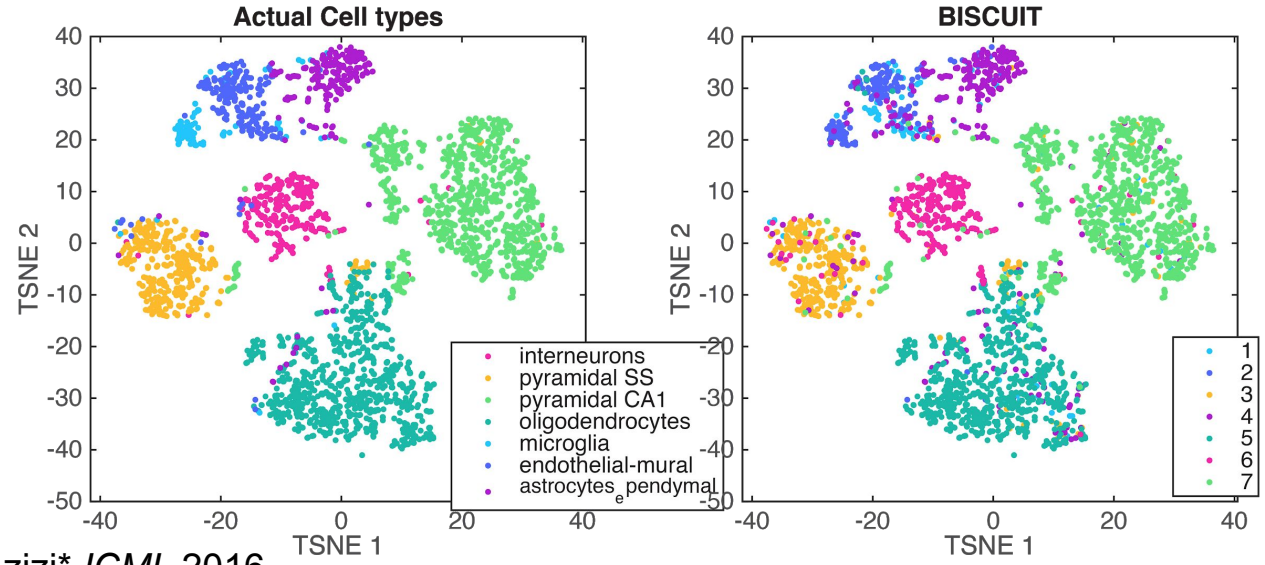
Prabhakaran*, Azizi* *ICML* 2016

Performance on Single-cell Data Zeisel et al., 2015

- 3005 mouse cortex cells, with UMIs
- Deep coverage gives good ground truth for **7 Cell types**
- Selected 558 genes with largest standard deviation across cells

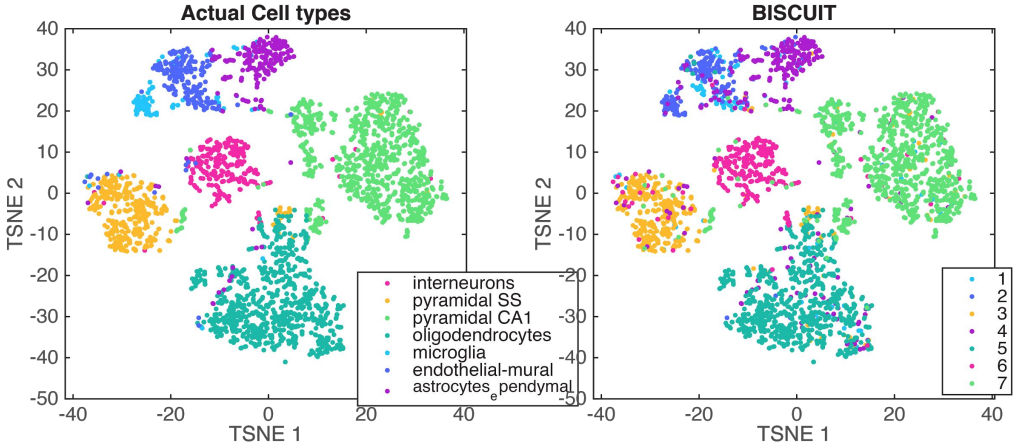
- Fit model to *log(counts+1)*

F-score: 0.91



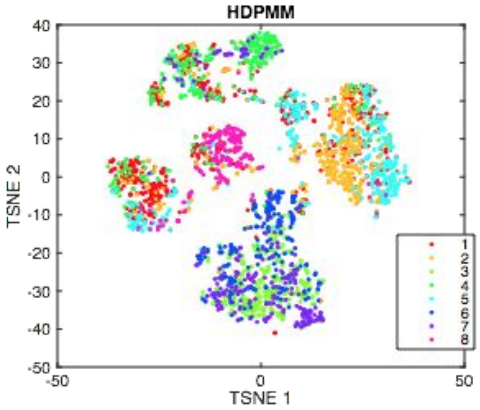
Prabhakaran*, Azizi* *ICML* 2016

Comparison to other methods

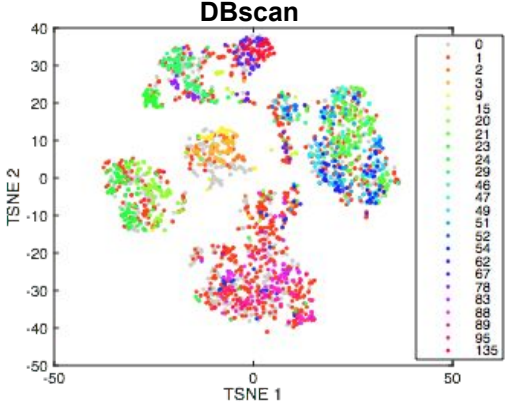


F-score: 0.91

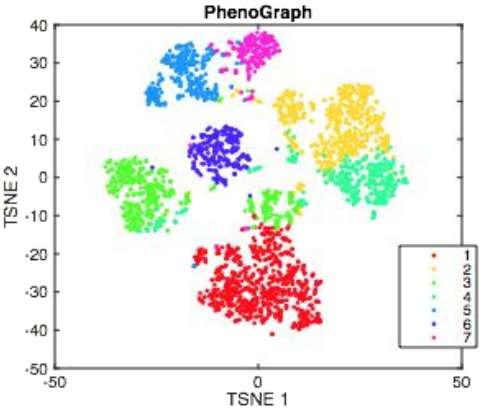
F-score: 0.79



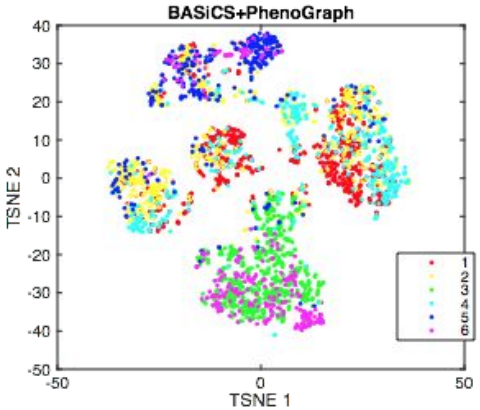
F-score: 0.21



F-score: 0.74



F-score: 0.61



Scaling BISCUIT for high-dimensional data: Challenges and solutions

Challenges to running BISCUIT

1. How do we achieve gene-gene covariance matrices that are highly ‘structured’?

2. Single-cell data ~ multivariate Gaussian distribution with prior / hyperprior distributions
   a. At least 5 gene-gene covariance matrix inversions per MCMC iteration
   b. Leads to time and space issues

3. MCMC Gibbs was performed in one giant sequential run
   a. Zeisel data of 3000 cells x 500 genes took ~ 12 hours for completion

Challenges to running BISCUIT and their Solutions

1. How do we achieve gene-gene covariance matrices that are highly ‘structured’?
   Solution: Use the Fiedler vector to partition gene space for better ordering of genes

2. Single-cell data ~ multivariate Gaussian distribution with prior / hyperprior distributions
   a. At least 5 gene-gene covariance matrix inversions per MCMC iteration
   b. Leads to time and space issues
   Solution: Use the Normal-inverse Wishart (NIW) distribution instead of the multivariate Gaussian

3. MCMC Gibbs was performed in one giant sequential run
   a. Zeisel data of 3000 cells x 500 genes took ~12 hours for completion
   Solution:
   - Parallelised the code using gene partitions dictated by the Fiedler vector
   - Creation of a cell-assignment based Confusion matrix
   - Recoded the MCMC engine using *logical subsetting* in R

Fiedler vector

- Concept used in Spectral graph partitioning, graph-cut theory
- **Graph** is defined as having G genes and E edges.
- **Graph partitioning** enables finding subsets of genes that are (fully) connected.
- **Spectral graph partitioning** is finding these subsets based on the EVD of the Laplacian.
- Steps
  - Compute Laplacian matrix, L, of the *covariance matrix* of X
    - L = D - Cov(X) where D = diag(column sum of Cov(X))
    - L, D, Cov(X) are of dimensions (genes x genes)
  - Perform EVD(L) → Set of eigenvalues of L := Laplacian spectrum
  - The 2nd smallest eigenvalue in the Laplacian spectrum := Fiedler eigenvalue/algebraic connectivity
  - Order the genes based on the Fiedler eigenvector

Why is the Fiedler vector interesting?

- Fiedler eigenvector is a vector that has length = number of genes.
- The number of 0s (or near-0 values) in the Fiedler eigenvector ≅
  - number of possible graphs partitions
  - number of possible gene partitions in the given gene space
- Gives an approximate ordering of genes that may *optimally* contribute to splitting the data into different clusters
  - Can be seen as Information-bottleneck (IB) optimisation
- Compute Fiedler vector on a High-performance cluster (HPC) or Cloud computing platform (eg. AWS)

Normal-inverse Wishart (NIW) distribution

- Given that X ~ multivariate Gaussian distribution
  - Posterior of **μ** and **Σ** are

$$f(\mu_k|\cdot) \sim \mathcal{N}(\mu_k^p, \Sigma_k^p), \quad f(\Sigma_k^{-1}|\cdot) \sim Wish(H, \sigma)$$

$$\mu_k^p = \Sigma_k^p(\Sigma'^{-1}\mu' + \Sigma_k^{-1}(\sum_j x_j/\beta_j))$$

$$\Sigma_k^p = (\Sigma'^{-1} + \Sigma_k^{-1}\sum_j \alpha_j^2/\beta_j)^{-1}$$

$$H = (H' + S_x)^{-1}, \quad \sigma = \sigma' + n_k$$

  - Many matrix inversions, time and space complexities involved.

- **Normal-inverse-Wishart distribution** (or **Gaussian-inverse-Wishart distribution**)
  - a multivariate four-parameter family of continuous probability distributions
    - unknown **μ**, **Σ**, κ, ρ
  - Conjugate prior of a multivariate Gaussian distribution
    - unknown **μ** and **Σ**

Normal-inverse Wishart (NIW) distribution

- Idea is to use the NIW prior instead, to infer unknown **μ** and **Σ** parameters *jointly*
- Heavily simplifies the inference equations needed for MCMC
- Posterior of **μ** and **Σ** are now:

$$f(\mu_k, \Sigma_k | X, \mu_0, \Lambda_0, \rho_0, \kappa_0, \alpha_j, \beta_j) \sim \text{NIW}(\mu_k, \Sigma_k | \mu', \Lambda', \rho', \kappa', \alpha_j, \beta_j)$$

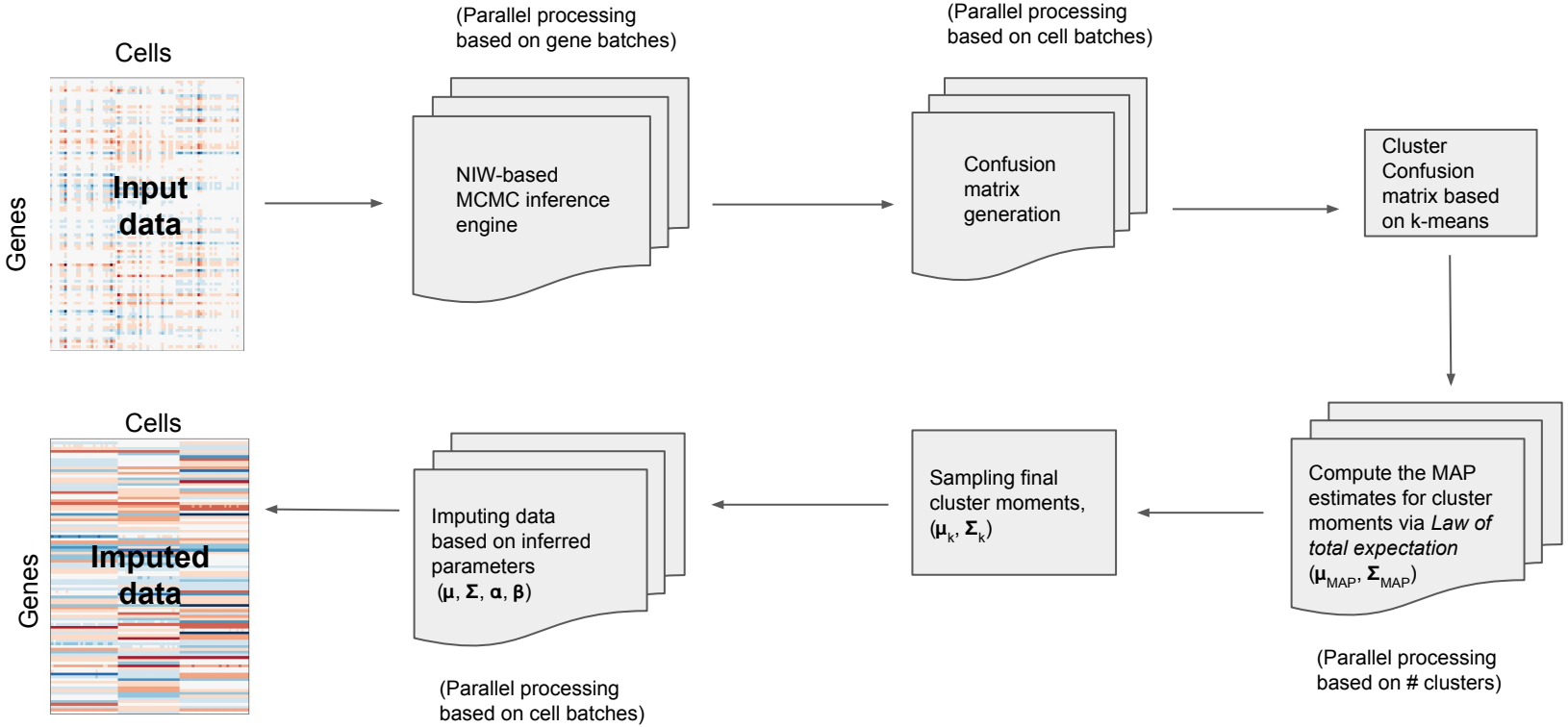
$$\mu' = \kappa_0 \mu_k + n_k, \kappa' = \kappa_0 + n, \rho' = \rho_o + n,$$

$$\Lambda' = \Lambda_0 + \sum_{j=1}^{N} (x_j - \bar{x})(x_j - \bar{x})^T + \kappa_0 \mu_0 \mu_0^T + \kappa' \mu' \mu'^T$$

$$f(\Sigma_k | X) \sim med(\beta_{j:z_j \in k}) * \text{Inverse Wish}(\Lambda'^{-1}, \rho')$$

$$f(\mu_k | X) \sim med(\alpha_{j:z_j \in k}) * \text{Student t}(\mu', \frac{\Lambda'}{\kappa'(\rho' - d + 1)})$$

Parallelizing schema



Code and relevant links

- Scalable BISCUIT:
  https://github.com/sandhya212/BISCUIT_SingleCell_IMM_ICML_2016

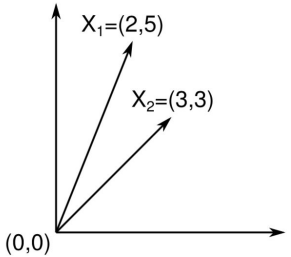
- NIW-based mixture model:
  https://github.com/sandhya212/Gibbs_IMM_MultivariateGaussian

- ICML paper and supplementary:
  http://proceedings.mlr.press/v48/prabhakaran16.html

Ideas to brainstorm for tomorrow or later

Idea 1: Understanding similarities and distances

X to S

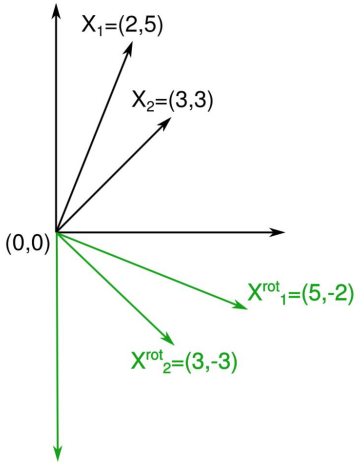


$S = XX^t$ ~ central Wishart

29	21
21	18

Idea 1: Understanding similarities and distances

X to S



$S = XX^t$ ~ central Wishart

29	21
21	18

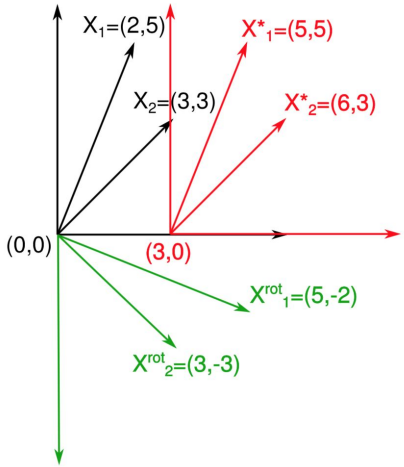
29	21
21	18

$S^{rot} = X^{rot}X^{rot^t}$ ~ central Wishart

X —Rotational information loss→ S

Idea 1: Understanding similarities and distances

X to S



$S = XX^t$ ~ central Wishart

29	21
21	18

$S^* = X^*X^{*t}$ ~ non-central Wishart

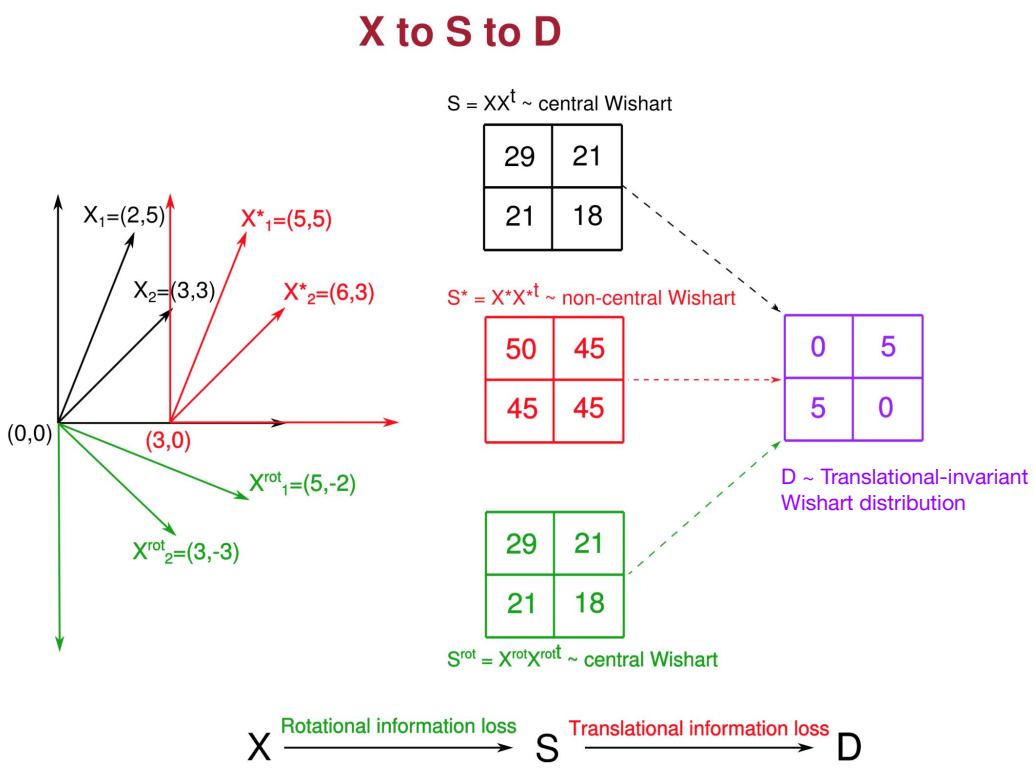
50	45
45	45

29	21
21	18

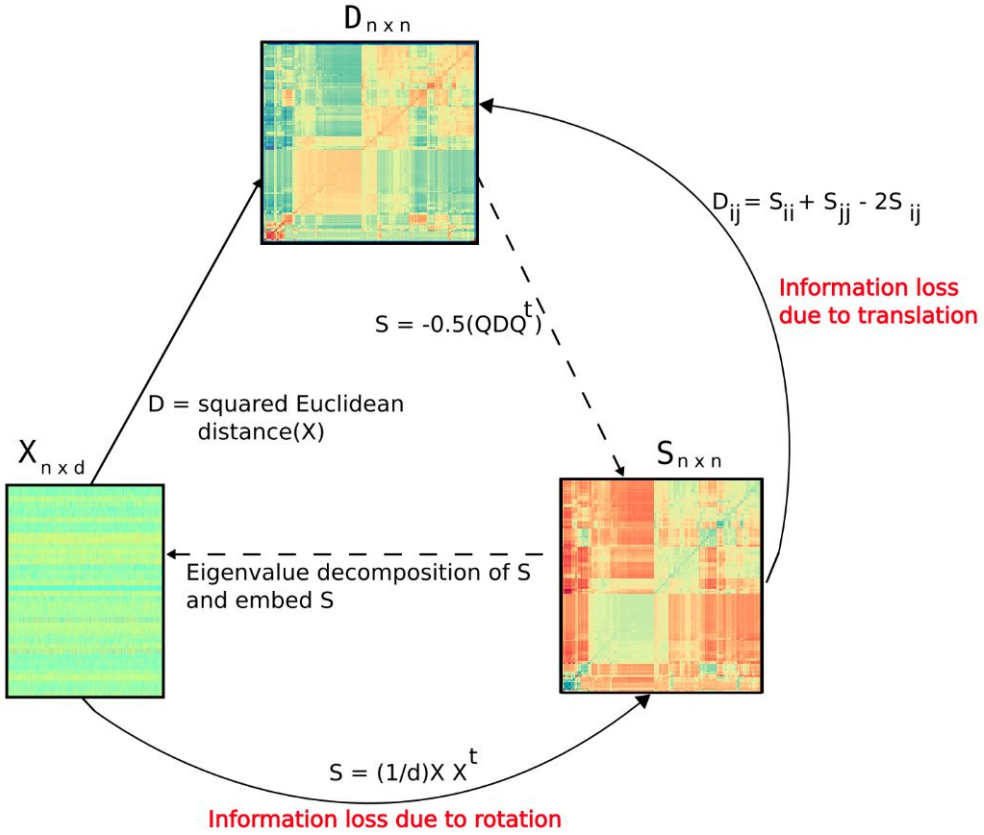
$S^{rot} = X^{rot}X^{rot^t}$ ~ central Wishart

X —Rotational information loss→ S

Idea 1: Understanding similarities and distances



Idea 1: Understanding similarities and distances



Idea 2: Distributed MCMC or Variational Bayes

- The gene space is high-dimensional
    - MCMC requires large number of iterations to reach equilibrium → time and computationally expensive
    - Variational Bayes gives an approximation to the posterior but there is no guarantee
- Idea is to run MCMC or VB independently on separate machines and then synthesize the posterior samples into a full distribution
    - Use MapReduce (Dean & Ghemawat, 2008)
- Advantages: no need to worry about parallelization of MCMC or VB that leads to an extra cluster alignment procedure

Idea 3: Autoencoder + Hamiltonian Monte Carlo (HMC)

- The gene space is high-dimensional
- Idea
  - Use the HMC to choose a decent subset of points in the high-dimensional space
  - Map the high-dimensional points to low-dimensions
    - Encoder part of an autoencoder
  - Explore the latent space to generate samples using another HMC
  - Map these samples back to the original space for inference
    - Decoder part of the auto encoder
    - Each sample can be treated as a Metropolis-Hastings proposal
- Advantage: a computationally efficient Bayesian sampling method for high-dimensional probabilities

Summary

- Analyzing single cell data involves computational challenges: dropouts, technical variation dependent on cell types
- BISCUIT:
  - A Bayesian approach for simultaneous clustering and imputing
  - Clusters identified with both mean and gene covariance patterns
  - Incorporating covariance information improves normalization and imputing
- Scalability:
  - Fiedler vector-based gene space partitioning
  - NIW distribution replacing the multivariate Gaussian
  - Parallelising the MCMC inference engine, confusion matrix computation and imputation
  - Zeisel data of 3000 cells x 500 genes takes ~ 45 mins for completion

Acknowledgements

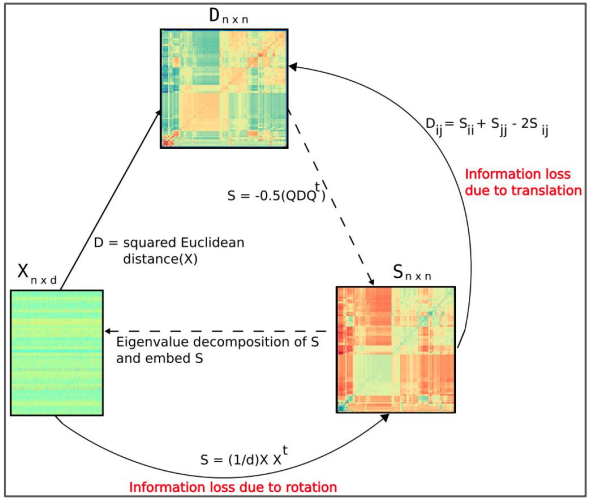
Dana Pe’er

Pe’er Lab
Elham Azizi
Ambrose Carr

Barbara Engelhardt (Princeton)
David Blei (Columbia)
Alp Kukucelbir (Columbia)

Leverage Bayesian methods for normalizing scRNA-seq data

- Idea 1: Use a translation-invariant covariance matrix to tide over translational and rotational biases in the data



- Idea 2: Distributed MCMC or Variational Bayes using MapReduce

- Idea 3: Autoencoder + HMC for sampling in low-dimensions

Thank you

@sandhya212