A Discussion on "A HIERARCHICAL BAYESIAN MODEL FOR SINGLE-CELL CLUSTERING USING RNA-SEQUENCING DATA"

Probabilistic Machine Learning Reading Group 07/22/2021

Roman Kouznetsov

University of Michigan Department of Statistics

UNIVERSITY OF MICHIGAN

Table of Contents

scRNA-seq Prerequisites

Q: What is sequencing?
A: The reading of arrangement of nucleotides to determine genetic expression.
Q: What is a transcript?
A: A (mRNA) transcript is the result of reading DNA to RNA (mRNA is of interest for us).
Q: Why do we need to use reverse transcription to recover genetic expression data? Why not go straight to the DNA?
A: Because DNA tells us the set of genes that exist in the cell, but mRNA tells us which genes actually get used to carry out cell functions.
Q: What is genetic expression?
A: Genetic expression (in laymans terms) is the level of activity a gene exhibits in a cell.
Q: How does it get measured?
A: After mRNA -> cDNA, the cDNA is copied a certain number of times, and the genes get counted until we reach a count equal to the

scRNA-seq Intro

Q: What is scRNA-seq?

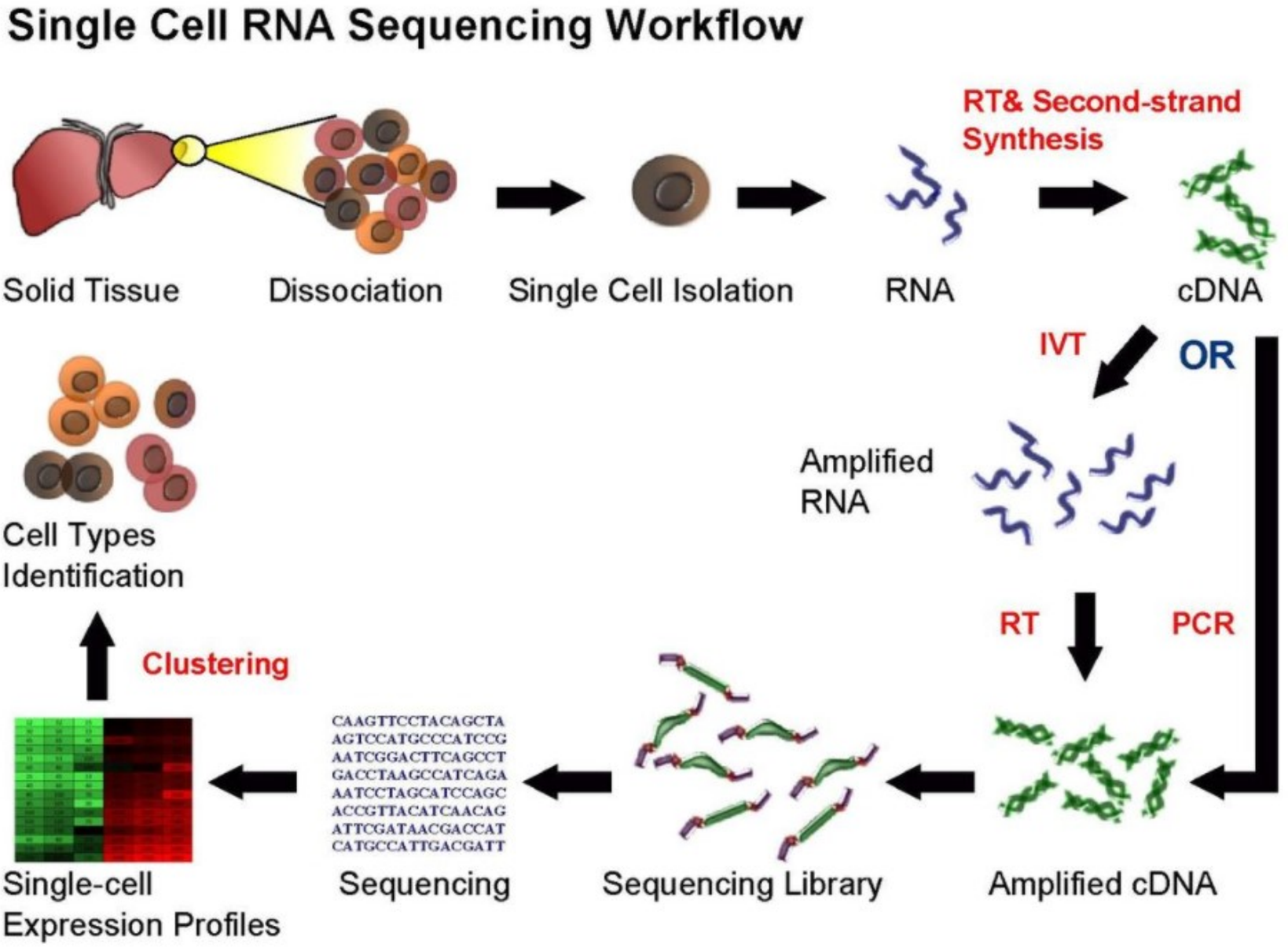
A: scRNA-seq is technology that allows us to read genetic expressions **at the cell level**.

Q: How does it get measured?

A: Reverse Transcription to cDNA to Amplification, Reading until Reach Library Size

Real A: You can treat this like a black box. Just trust that the result is genetic expression of various genes.

(but why?) scRNA-seq Intro



Data (Hey, that's us!)

The data outputted by this entire process is a matrix of cells by genes (CxG) where each row represents a cell's genetic expressions of all G genes.

Questions Learned from Data

Q: What can be learned from this data?

A: **Clustering of cells.** "Example: This can uncover the existence of rare cell types within a cell population that may never have been seen before. For example, rare specialized cells in the lung called pulmonary ionocytes that express the Cystic Fibrosis Transmembrane Conductance Regulator were identified in 2018 by two groups performing scRNA-Seq on lung airway epithelia." - (Montoro DT, Haber AL, Biton M, Vinarsky V, Lin B, Birket SE, et al.)

CFTCR is a membrane protein which lets specific ions into the lung! So these groups identified a cell type in the lung that allows for ions to travel into lung cells. This refines our understanding of what cells do what in the lung! This clustering procedure could help with disease prevention and cell function understanding!

So... what is this about then?

This paper addresses 2 tasks:

1. Can a DPMM be constructed to classify cells into clusters?
2. Can using this information about clusters help us determine the probability that a gene's NA representation is accurate?

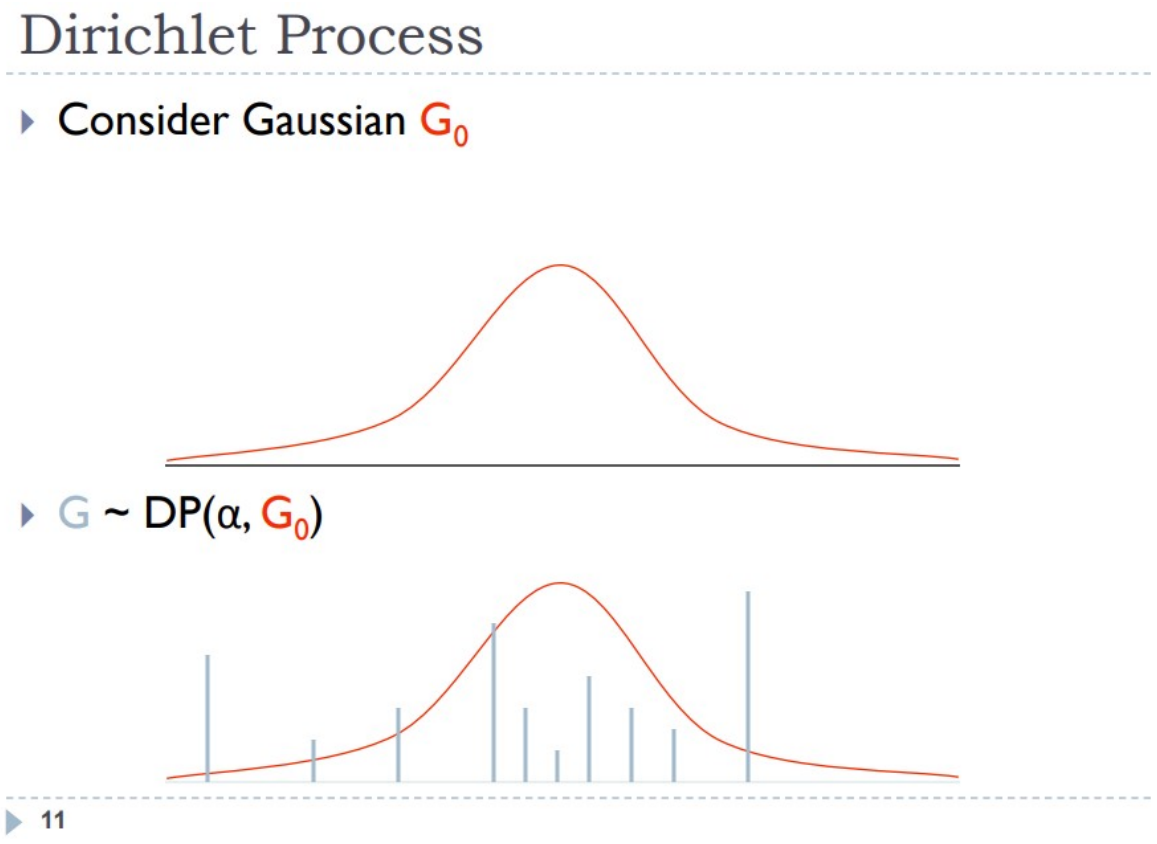
What is a Dirichlet Process?

A Dirichlet Process (DP) is a stochastic process used to cluster groups of data when the number of clusters is unknown, which cannot be said about

- GMMs
- K-Means
- Agglomerative Clustering
- SIMLR and ZIFA (Competing Methods in this paper)

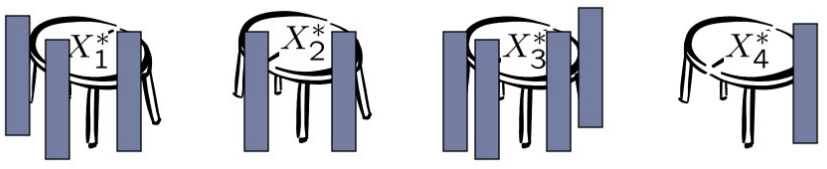
Visualization

A Dirichlet distribution takes a base distribution G_0 and a parameter α which acts as a propensity to take new samples from the base distribution. The end result is shown before you.



DP as Chinese Restaurant Process

Chinese Restaurant Process



Chinese Restaurant Process

$$X_n | X_1, \ldots, X_{n-1} = \begin{cases} X_k^* & \text{with probability } \frac{\text{num}_{n-1}(X_k^*)}{n-1+\alpha} \\ \text{new draw from } G_0 & \text{with probability } \frac{\alpha}{n-1+\alpha} \end{cases}$$

Consider a restaurant with infinitely many tables, where the X_n's represent the patrons of the restaurant. From the above conditional probability distribution, we can see that a customer is more likely to sit at a table if there are already many people sitting there. However, with probability proportional to α, the customer will sit at a new table.

Dirichlet Process: Pros and Cons

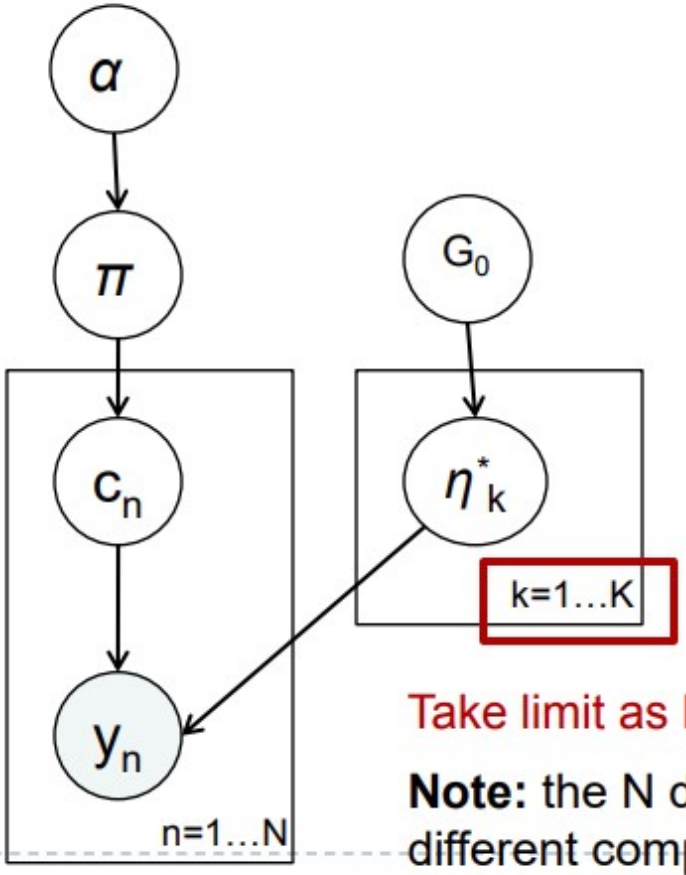
- Pros:
  - No need to specify K beforehand.
  - Clusters are based on probabilistic metrics (not distance or predetermined metric).
  - Can learn mixture distributions quite well.
  - Fairly easy to implement.
- Cons:
  - Easy sampling requires some (semi)conjugacy assumptions that may not always be appropriate.
  - Base distribution needs to be specified.
  - Clusters don't always have an interpretable meaning.

Clustering Inference with Dirichlet Process

The model is trained with MCMC and the cluster labels are a part of this process. So, after some burn-in removal and thinning, we have posterior samples of the cluster label.

Representation of DP as a Mixture Model

- An infinite mixture model assumes that the data come from a mixture of an *infinite* number of distributions



$\pi \sim \text{Dirichlet}(\alpha/K, \ldots, \alpha/K)$

$c_n \sim \text{Multinomial}(\pi)$

$\eta_k \sim G_0$

$y_n \mid c_n, \eta_1, \ldots \eta_K \sim F(\cdot \mid \eta_{c_n})$

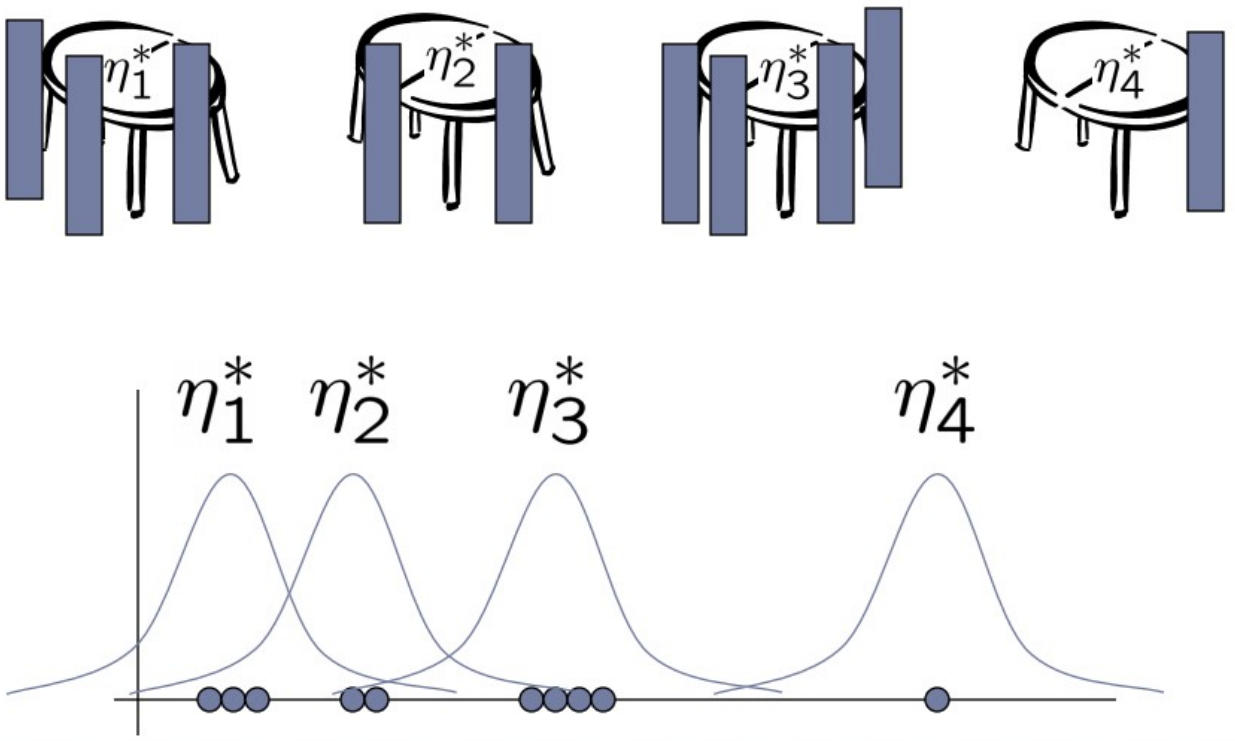
Take limit as K goes to ∞

Note: the N data points still come from at most N different components

[Rasmussen 2000]

CRP as Mixture

CRP Mixture



Prior Specification

$$p\left(\widetilde{\pi} \mid \boldsymbol{c}, \alpha_{\pi}, \beta_{\pi}\right) = \prod_{k=1}^{K} \prod_{j=1}^{G} \frac{\Gamma(\alpha_{\pi}+\beta_{\pi})}{\Gamma(\alpha_{\pi})\Gamma(\beta_{\pi})} \tilde{\pi}_{kj}^{\alpha_{\pi}-1} \left(1-\tilde{\pi}_{kj}\right)^{\beta_{\pi}-1}$$

Likelihood Specification

$$p(\boldsymbol{z} \mid \boldsymbol{c}, \widetilde{\boldsymbol{\pi}}) = \prod_{k=1}^{K} \prod_{i:c_i=k} \prod_{j=1}^{G} \tilde{\pi}_{kj}^{z_{ij}} \left(1 - \tilde{\pi}_{kj}\right)^{1-z_{ij}}$$

Posterior Specification

Notice that because the prior was Beta and the likelihood was Bernoulli, we actually have an analytic posterior we can sample from due to conjugacy! This analytic posterior makes it easy to integrate out π and get a full conditional posterior for z INDEPENDENT of π!!!

Formal Model

- $z_{ij} | \pi_{ij} \sim Bernoulli(\pi_{ij})$
- $(\pi_{i1}, \pi_{i2}, \pi_{i3}, \ldots, \pi_{iG}) \sim DP(\alpha_\lambda, G_0)$
- $y_{ij} \mid z_{ij}, g_j, \delta_j, \sigma^2_{0j}, \sigma^2_{1j} \sim \mathrm{N}\left(g_j + \delta_j z_{ij} + s_i, \sigma^2_{0j}\left(1 - z_{ij}\right) + \sigma^2_{1j} z_{ij}\right)$
  - g_j: the sum of $g_j^{'}$, log gene length, and a log scaling constant
  - s_i: log library size
- $$x_{ij} \mid y_{ij} = \begin{cases} y_{ij} & \text{w.p. } 1 - \Phi\left(\gamma_1 y_{ij} + \gamma_2\right) \\ \text{N.A.} & \text{w.p. } \Phi\left(\gamma_1 y_{ij} + \gamma_2\right) \end{cases}$$

Formal Model pt. 2

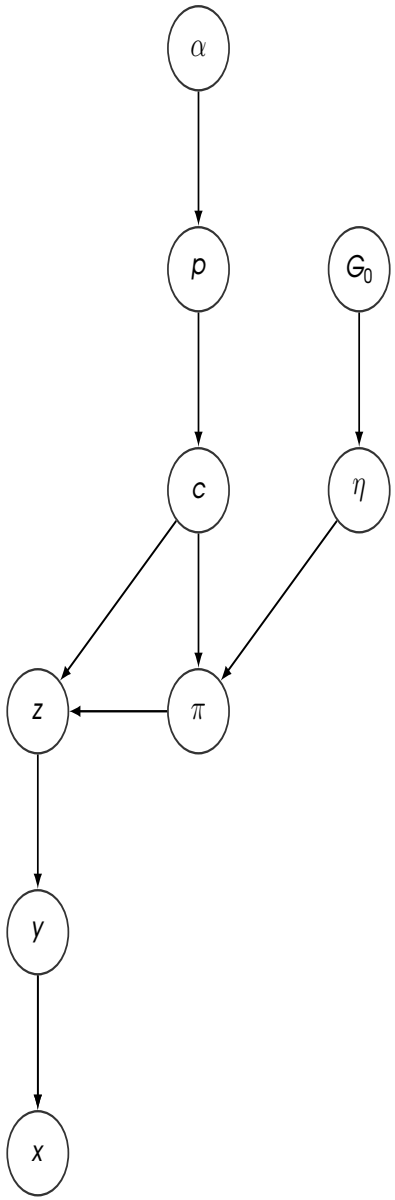


Figure: Bayesian Graph of Sampling from Dirichlet Process

The (Brutal) MCMC Algorithm

S1. Sampling algorithm. We provide the details of the MCMC sampling algorithm used in our method. Let $\boldsymbol{\Theta}$ be the set of all parameters after integrating away $\boldsymbol{\pi}$ and $y_{ij}(x_{ij} = \text{N.A.})$. Suppose we have K components in DP part of the model in a particular iteration, we have posterior:

$$
\begin{aligned}
p(\boldsymbol{\Theta}|\boldsymbol{x}) \;=\; & \prod_{i=1}^{N}\prod_{j=1}^{G}\Phi\Big(\frac{\gamma_1(g_j+\delta_j z_{ij}+s_i)+\gamma_2}{\sqrt{1+\gamma_1^2(\sigma_{0j}^2(1-z_{ij})+\sigma_{1j}^2 z_{ij})}}\Big)^{1\{x_{ij}=N.A.\}} \\
& \cdot \Big\{\frac{1-\Phi(\gamma_1 x_{ij}+\gamma_2)}{\sqrt{2\pi(\sigma_{0j}^2(1-z_{ij})+\sigma_{1j}^2 z_{ij})}}\exp\Big[-\frac{(x_{ij}-g_j-\delta_j z_{ij}-s_i)^2}{2(\sigma_{0j}^2(1-z_{ij})+\sigma_{1j}^2 z_{ij})}\Big]\Big\}^{1\{x_{ij}\neq N.A.\}} \\
& \times \prod_{j=1}^{G}\frac{1}{\sqrt{2\pi\tau_g^2}}\exp\Big[-\frac{(g_j-\nu_g)^2}{2\tau_g^2}\Big]\times\prod_{j=1}^{G}\frac{\beta_\delta^{\alpha_\delta}}{\Gamma(\alpha_\delta)}\delta_j^{\alpha_\delta-1}\exp(-\beta_\delta\delta_j) \\
& \times \prod_{j=1}^{G}\frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}(\sigma_{0j}^2)^{-\alpha_0-1}\exp\Big(-\frac{\beta_0}{\sigma_{0j}^2}\Big)\cdot\frac{\beta_1^{\alpha_1}}{\Gamma(\alpha_1)}(\sigma_{1j}^2)^{-\alpha_1-1}\exp\Big(-\frac{\beta_1}{\sigma_{1j}^2}\Big) \\
& \times \prod_{k=1}^{K}\prod_{j=1}^{G}\frac{\Gamma(\alpha_\pi+\beta_\pi)}{\Gamma(\alpha_\pi)\Gamma(\beta_\pi)}\cdot\frac{\Gamma(\alpha_\pi+\sum_{i:c_i=k}z_{ij})\Gamma(\beta_\pi+n_k-\sum_{i:c_i=k}z_{ij})}{\Gamma(\alpha_\pi+\beta_\pi+n_k)} \\
& \times \frac{\Gamma(\alpha_\lambda)\alpha_\lambda^K\prod_{k=1}^{K}(n_k-1)!}{\Gamma(\alpha_\lambda+n)}\times\frac{b_l^{a_l}}{\Gamma(a_l)}\alpha_\lambda^{a_l-1}\exp(-b_l\alpha_\lambda) \\
& \times \frac{1}{\sqrt{2\pi\tau_1^2}}\exp\Big(-\frac{(\gamma_1-\nu_1)^2}{2\tau_1^2}\Big)\times\frac{1}{\sqrt{2\pi\tau_2^2}}\exp\Big(-\frac{(\gamma_2-\nu_2)^2}{2\tau_2^2}\Big) \\
& \times \frac{1}{\sqrt{2\pi d_g^2}}\exp\Big(-\frac{(\nu_g-m_g)^2}{2d_g^2}\Big)\times\frac{b_g^{a_g}}{\Gamma(a_g)}(\tau_g^2)^{-a_g-1}\exp\Big(-\frac{b_g}{\tau_g^2}\Big) \\
& \times \frac{b_d^{a_d}}{\Gamma(a_d)}\beta_\delta^{a_d-1}\exp(-b_d\beta_\delta)\times\frac{b_d^{a_d}}{\Gamma(a_d)}\beta_0^{a_d-1}\exp(-b_d\beta_0) \\
& \times \frac{b_d^{a_d}}{\Gamma(a_d)}\beta_1^{a_d-1}\exp(-b_d\beta_1)\times\frac{b_p^{a_p}}{\Gamma(a_p)}\beta_\pi^{a_\pi-1}\exp(-b_\pi\beta_\pi).
\end{aligned}
$$

Output of Model Training

What does the model actually give us to work with?
For each iteration that the Markov chain runs, we get a new value for $\mathbf{c}$, so we have posterior estimates for the cluster of each cell.
Additionally, we have posterior estimates for all parameters in the model, giving us posterior samples of parameters of interest that define important functions for us.

Outline

Before running MCMC.

1. Integrate out parameters that aren't really of interest to us (π and y_{ij}).
2. Get the full conditional posteriors of all parameters.

For $m = 1, 2, \ldots M$

1. Sample binary expression variable: $z_{ij} \sim Bernoulli(p^*_{ij})$
2. Sample from the full conditional posterior of c_i for $i = 1, 2, \ldots N$. **Note that this has the affect of removing a class from the next iteration if c_i is the only label with that class.**
3. Take Gibbs updates for all other parameters in the model.

How do we actually determine the cluster assignment?

1. **Approach 1** The matrix below
   $S_{ii'} = \frac{1}{M} \sum_{m=1}^{M} \delta\left(c_i^{(m)}, c_{i'}^{(m)}\right), \quad i, i' = 1, 2, \ldots, N$
   gets plugged in to an ARI objective function and the partition of **c** that maximizes the objective is the selected clustering.
2. **Approach 2** Hierarchical clustering based on the **z** vector for each cell.

Evaluation of Cell Clusters

In order to evaluate the performance of the clustering model, we need a metric that accurately summarizes how "different" an entire cluster assignment is from the ground truth.

Enter: ARI and NMI

Dissecting ARI

The next few slides might be insulting to your intelligence, but it will hopefully clarify what this mess means.

Let U_p be a partition of the data into p classes and V_q be the ground truth about said data that is separated into q classes.

$$\begin{aligned}
&\operatorname{ARI}(U, V) \\
&= \frac{\sum_{p=1}^{P}\sum_{q=1}^{Q}\binom{|U_p \cap V_q|}{2} - \sum_{p=1}^{P}\binom{|U_p|}{2}\sum_{q=1}^{Q}\binom{|V_q|}{2}\Big/\binom{n}{2}}{\left[\sum_{p=1}^{P}\binom{|U_p|}{2} + \sum_{q=1}^{Q}\binom{|V_q|}{2}\right]/2 - \sum_{p=1}^{P}\binom{|U_p|}{2}\sum_{q=1}^{Q}\binom{|V_q|}{2}\Big/\binom{n}{2}}
\end{aligned}$$

Dissecting ARI (cont.)

Say I have a contingency table that looks like this:

	Gene 1	Gene 2	Total
Cell 1	30	10	40
Cell 2	15	25	40
Cell 3	15	5	20
Total	60	40	100

Dissecting ARI (cont.)

Say I have a contingency table that looks like this:

	Gene 1	Gene 2	Total
Cell 1	30	10	40
Cell 2	15	25	40
Cell 3	15	5	20
Total	60	40	100

If I were to make a prediction of expected value in each cell assuming no prior knowledge about the cells, I would just take the expected count for each cell.
For example, (1,1) would be 60*(40/100) = 24, (1,2) would be 40*(40/100) = 16, and so on.

Dissecting ARI (cont.)

$$\mathrm{ARI}(U, V)$$

$$= \frac{\sum_{p=1}^{P}\sum_{q=1}^{Q}\binom{|U_p \cap V_q|}{2} - \sum_{p=1}^{P}\binom{|U_p|}{2}\sum_{q=1}^{Q}\binom{|V_q|}{2}\Big/\binom{n}{2}}{\left[\sum_{p=1}^{P}\binom{|U_p|}{2} + \sum_{q=1}^{Q}\binom{|V_q|}{2}\right]/2 - \sum_{p=1}^{P}\binom{|U_p|}{2}\sum_{q=1}^{Q}\binom{|V_q|}{2}\Big/\binom{n}{2}}$$

The expected number of pairs that line up between 2 partitions under expected assignment is exactly the second term in both the numerator and denominator!

Dissecting ARI (cont.)

$$\mathrm{ARI}(U, V)$$

$$= \frac{\sum_{p=1}^{P}\sum_{q=1}^{Q}\binom{|U_p \cap V_q|}{2} - \sum_{p=1}^{P}\binom{|U_p|}{2}\sum_{q=1}^{Q}\binom{|V_q|}{2}\Big/\binom{n}{2}}{\left[\sum_{p=1}^{P}\binom{|U_p|}{2} + \sum_{q=1}^{Q}\binom{|V_q|}{2}\right]/2 - \sum_{p=1}^{P}\binom{|U_p|}{2}\sum_{q=1}^{Q}\binom{|V_q|}{2}\Big/\binom{n}{2}}$$

The first term in numerator is the total number of pairs that exist in the same cluster in both partitions.

The first term in the denominator is the best case scenario where every countable pair in each cluster is correct.

So, what we have in essence is $\frac{\text{correct pairs - expected correct pairs}}{\text{max possible correct pairs - expected correct pairs}}$

ARI

Ultimately, the ARI is a value that tries to assess the proportion of pairs that are clustered better than at random out of the maximum possible pairs better than random assignment.

NMI

A

$$
\begin{aligned}
\mathrm{MI}(U, V) &= \sum_{p=1}^{P}\sum_{q=1}^{Q} \frac{|U_p \cap V_q|}{N} \log \frac{N|U_p \cap V_q|}{|U_p| \times |V_q|} \\
&= \sum_{p=1}^{P}\sum_{q=1}^{Q} P(U_p \cap V_q) log \left(\frac{N * N * P(U_p \cap V_q)}{N * P(U_p)) * N * P(V_q)} \right) \\
&= \sum_{p=1}^{P}\sum_{q=1}^{Q} P(U_p \cap V_q) log \left(\frac{P(U_p \cap V_q)}{P(U_p) * P(V_q)} \right) \\
&= \sum_{p=1}^{P}\sum_{q=1}^{Q} P(U_p \cap V_q) log \left(\frac{P(U_p \cap V_q)}{P(U_p)} \right) - \sum_{p=1}^{P}\sum_{q=1}^{Q} P(U_p \cap V_q) log(P(V_q)) \\
&= \sum_{p=1}^{P} P(U_p) \sum_{q=1}^{Q} P(V_q|U_p = p) * log(P(V_q|U_p = p)) - \sum_{q=1}^{Q} log(P(V_q)) \sum_{p=1}^{P} P(U_p \cap V_q) \\
&= -\sum_{p=1}^{P} P(U_p) H(V_q|U_p = p) - \sum_{q=1}^{Q} P(V_q)) log(P(V_q)) = -H(V_q|U_p) - (-H(V_q)) \\
&= H(V_q) - H(V_q|U_p)
\end{aligned}
$$

In other words, if V_q is the truth and U_p is the clustering, the mutual information is the drop in entropy once we know the cluster assignment. We want it to be as large as possible, meaning that the ground truth is perfectly recovered with no uncertainty.

ARI, NMI Summary

- ARI closer to 1 is better.
- NMI closer to 1 is better.
- ARI focuses on the proportion of possible accuracy over complete randomness while NMI focuses on reduction of entropy.

Competing Methods

1. BackSPIN: A clustering algorithm recursively splits cells into group to maximize some with-group correlation. *Requires user to specify depth.*
2. CIDR: Learn dropout parameter through logistic regression and use that to calculate dissimilarity matrix for cells. Perform PCA and perform hierarchical clustering. *Can operate with K known and unknown.*
3. SIMLR: Learn a cell-cell similarity metric and use a stochastic neighbor embedding method to project to latent space for clustering. *Requires use to specify K and latent space dimension.* (ZIFA operates similarly but with factor analysis as dimension reduction technique.)
4. SNN-Cliq: Calculate graph partition based on shared nearest neighbors from Euclidean distance between cell genetic expressions. *Clustering requires control over 2 parameters.*

Experiment Objectives

1. Cluster well even when facing dropout.
2. Learn decent posterior means of the model parameters.

Simulation Study

The simulation study does the following:

1. Simulate 300 cells, each with 200 genes (N=300, G=200)
2. For each cell assign them to one of 10 clusters with equal probability,
3. For each class, sample $\pi_k \sim Beta(0.9, 0.9)$ a G-dim vector describing the expression probability for each gene.
4. Sample $z \sim Bernoulli(\pi_k)$ for each cell based on its class.
5. Plug all the parameters into the model using...
   - $g_j \sim N(-23, 9)$
   - $s_i \sim N(20, 4)$
   - $\delta_j \sim TruncNormal(5 or 8, 4)$
   - $\sigma_{0j}^2 \sim TruncNormal(6, 4)$
   - $\sigma_{1j}^2 \sim TruncNormal(6, 4)$

The goal is to recover all the parameters in 4 and the cluster class for each cell!

How do we assess the results?

Methods are compared based on their NMI and ARI.

TABLE S1

ARI of BasClu, CIDR, SIMLR, and ZIFA in the simulation study.

Scenario	**1**	**2**	**3**	**4**
BasCluZ	0.958 (0.030)	1.000 (0.000)	0.995 (0.004)	0.999 (0.003)
SIMLR	0.102 (0.059)	0.385 (0.132)	0.674 (0.131)	0.841 (0.055)
ZIFA (5)	0.297 (0.112)	0.489 (0.046)	0.564 (0.041)	0.673 (0.056)
Scenario	**5**	**6**	**7**	**8**
BasCluZ	0.999 (0.003)	1.000 (0.000)	0.999 (0.003)	0.999 (0.003)
SIMLR	0.963 (0.020)	0.926 (0.031)	0.944 (0.015)	0.956 (0.022)
ZIFA (5)	0.746 (0.053)	0.766 (0.061)	0.702 (0.070)	0.717 (0.086)
Scenario	**9**	**10**	**11**	**12**
BasCluZ	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
SIMLR	0.676 (0.111)	0.932 (0.036)	0.977 (0.026)	0.969 (0.027)
ZIFA (5)	0.606 (0.103)	0.731 (0.073)	0.824 (0.070)	0.892 (0.058)
Scenario	**13**	**14**	**15**	**16**
BasCluZ	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
SIMLR	0.988 (0.027)	1.000 (0.000)	0.969 (0.033)	0.989 (0.016)
ZIFA (5)	0.903 (0.035)	0.900 (0.038)	0.881 (0.067)	0.916 (0.039)

The number in each cell is the mean value across the 5 replicates, with stand
deviation in the parentheses. In the parentheses following "ZIFA", the value den
pre-specified number of latent factors.

(a) Results Based on ARI Evaluation

NMI of BasClu, CIDR, SIMLR and ZIFA in the simulation study

Scenario	1	2	3	4
BasCluZ	0.965 (0.025)	1.000 (0.000)	0.996 (0.004)	0.999 (0.003)
CIDR	0.182 (0.046)	0.288 (0.070)	0.361 (0.059)	0.543 (0.081)
SIMLR	0.271 (0.084)	0.552 (0.115)	0.758 (0.089)	0.888 (0.044)
ZIFA (5)	0.438 (0.085)	0.621 (0.050)	0.680 (0.022)	0.767 (0.037)
Scenario	5	6	7	8
BasCluZ	0.999 (0.003)	1.000 (0.000)	0.999 (0.003)	0.999 (0.003)
CIDR	0.670 (0.071)	0.652 (0.069)	0.647 (0.073)	0.654 (0.086)
SIMLR	0.970 (0.014)	0.952 (0.019)	0.959 (0.006)	0.966 (0.016)
ZIFA (5)	0.812 (0.045)	0.826 (0.046)	0.785 (0.052)	0.795 (0.066)
Scenario	9	10	11	12
BasCluZ	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
CIDR	0.289 (0.027)	0.508 (0.065)	0.797 (0.047)	0.909 (0.030)
SIMLR	0.772 (0.064)	0.956 (0.019)	0.987 (0.013)	0.985 (0.013)
ZIFA (5)	0.711 (0.064)	0.817 (0.037)	0.881 (0.041)	0.920 (0.037)
Scenario	13	14	15	16
BasCluZ	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
CIDR	0.952 (0.014)	0.958 (0.015)	0.951 (0.028)	0.949 (0.032)
SIMLR	0.995 (0.011)	1.000 (0.000)	0.984 (0.016)	0.995 (0.007)
ZIFA (5)	0.930 (0.021)	0.926 (0.023)	0.921 (0.046)	0.937 (0.024)

(b) Results Based on NMI Evaluation

Applied Study

BasCluZ (CLARIFY WHAT THIS MEANS) is put to the test against competing methods on 3 different data sets that have different values of *K* as its "ground truth."

1. Okaty (K = 6)
2. Pollen (K = 8)
3. Usoskin (K = 11)

Experiment Results

TABLE S5
ARI on three scRNA-seq datasets.

Dataset	BasCluS*	BasCluZ	BackSPIN (2)*	BackSPIN (2, all)*	BackSPIN (3)*	BackSPIN (3, all)*
Okaty	0.673 [10]	0.566	0.288 [4]	0.526 [4]	0.455 [8]	0.369 [8]
Pollen#	0.665 [19]	0.634	-	-	0.438 [8]	0.749 [8]
Usoskin	0.559 [12]	0.569	-	-	0.509 [7]	0.435 [8]
Dataset	BackSPIN (4)*	BackSPIN (4, all)*	CIDR0*	CIDR0 (all)*	CIDR	CIDR (all)
Okaty	-	-	0.358 [8]	0.276 [6]	0.390	0.276
Pollen#	-	-	0.356 [6]	0.405 [6]	0.383	0.389
Usoskin	0.409 [13]	0.318 [16]	0.734 [4]	0.777 [4]	0.558	0.386
Dataset	SIMLR	SIMLR (all)	SNN-Cliq (0.3)*	SNN-Cliq (0.3, all)*	SNN-Cliq (0.4)*	SNN-Cliq (0.4, all)*
Okaty	0.521	0.474	0.432 [6]	0.108 [3]	0.535 [8]	0.108 [3]
Pollen#	0.545	0.777	0.192 [7]	0.010 [3]	0.411 [13]	0.332 [13]
Usoskin	0.452	0.413	0.519 [10]	0.045 [3]	0.657 [17]	0.265 [9]
Dataset	SNN-Cliq (0.5)*	SNN-Cliq (0.5, all)*	ZIFA (5)	ZIFA (5, all)	ZIFA (10)	ZIFA (10, all)
Okaty	0.544 [7]	0.282 [4]	0.435	0.118	0.110	0.034
Pollen#	0.383 [18]	0.245 [17]	0.597	0.469	0.388	0.336
Usoskin	0.167 [61]	0.261 [20]	0.515	0.203	0.442	0.160
Dataset	ZIFA (20)	ZIFA (20, all)	rawZ	rawZ (all)	MASTZ	MASTZ (all)
Okaty	0.049	0.058	0.606	0.367	0.301	0.039
Pollen#	0.322	0.399	0.407	0.137	0.127	0.116
Usoskin	0.271	0.059	0.449	0.084	0.694	0.134

(a) Results Based on ARI Evaluation

NMI on three scRNA-seq datasets

Dataset	BasCluS*	BasCluZ	BackSPIN (2)*	BackSPIN (2, all)*	BackSPIN (3)*	BackSPIN (3, all)*
Okaty	0.740 [10]	0.713	0.431 [4]	0.573 [4]	0.604 [8]	0.567 [8]
Pollen#	0.693 [19]	0.723	–	–	0.590 [8]	0.795 [8]
Usoskin	0.689 [12]	0.696	–	–	0.703 [7]	0.647 [8]
Dataset	BackSPIN (4)*	BackSPIN (4, all)*	CIDR0*	CIDR0 (all)*	CIDR	CIDR (all)
Okaty	–	–	0.611 [8]	0.478 [6]	0.604	0.478
Pollen#	–	–	0.561 [6]	0.525 [6]	0.584	0.532
Usoskin	0.606 [13]	0.543 [16]	0.594 [4]	0.643 [4]	0.643	0.568
Dataset	SIMLR	SIMLR (all)	SNN-Cliq (0.3)*	SNN-Cliq (0.3, all)*	SNN-Cliq (0.4)*	SNN-Cliq (0.4, all)*
Okaty	0.700	0.661	0.613 [6]	0.287 [3]	0.656 [8]	0.287 [3]
Pollen#	0.682	0.846	0.354 [7]	0.065 [3]	0.566 [13]	0.526 [13]
Usoskin	0.641	0.588	0.499 [10]	0.068 [3]	0.675 [17]	0.336 [9]
Dataset	SNN-Cliq (0.5)*	SNN-Cliq (0.5, all)*	ZIFA (5)	ZIFA (5, all)	ZIFA (10)	ZIFA (10, all)
Okaty	0.668 [7]	0.475 [4]	0.657	0.266	0.363	0.121
Pollen#	0.549 [18]	0.497 [17]	0.704	0.660	0.469	0.489
Usoskin	0.437 [61]	0.428 [20]	0.630	0.434	0.552	0.373
Dataset	ZIFA (20)	ZIFA (20, all)	rawZ	rawZ (all)	MASTZ	MASTZ (all)
Okaty	0.197	0.144	0 737	0.506	0.501	0.141
Pollen#	0.436	0.494	0 570	0.365	0.354	0.246
Usoskin	0.450	0.089	0 527	0.244	0.702	0.337

(b) Results Based on NMI Evaluation

Major Takeaways

1. Clustering cells into types is a statistical problem that has actually yielded solutions desired by scientists.
2. The Dirichlet Process method performs better almost uniformly in comparison to competing methods, EVEN in the face of dropout.
3. The DP is able to do very well partly due to the fact that they do not try to estimate π which is insanely large in dimension. This narrows the parameter space and makes the inference reasonably fast.

Lingering Questions

1. What ensures that the clustering process is helpful for learning the other parameters? Is it just me or would it have been interesting to see the results of learning dropout parameters WITHOUT clusters?
2. Couldn't the measured expression level itself had priors for a linear regression instead of just assuming that $y_{ij} = y^*_{ij} + s_i + l_j + \mu$?
3. What happens to all of these methods as we push K to either 1 or 100? I bet that example is impractical, but it would be nice to know if these methods can hold up if cells of the same "type" are passed into the model.