A method for cell-type deconvolution based on a sequential Monte Carlo framework

Kyle Coleman

9 May 2021

Reproducibility

The following code chunks can be run to reproduce the results show in Figure 1 of this paper, without and with parallelization. X is a matrix containing the cell-type specific gene expression (GE) levels for brain and heart cells, and the function hvg_selection() is used to reduce the dimensionality of X by selection the top (100*p)% highly variable genes, where p = 0.05 by default. We then load the matrix M that contains the proportion of cell types in the heterogeneous samples, as specified by the authours. The matrix Y, consisting of the GE levels of the heterogeneous samples, is then constructed by multiplying the matrices X and M. We then run the sequential Monte Carlo deconvolution algorithm, using the function smc_deconvolution(), to estimate the proportion of cell types in each of the samples of Y. K represents the number of cell types in the samples of Y, and L is the number of iterations of the algorithm. The parameter 'parallel' is set to FALSE by default. Along with the estimated cell type proportions, the algorithm also provides the runtime for each of the iterations if parallelization is not used, and the total runtime of the algorithm if parallelization is used.

No parallelization:

```r
library(smc.deconvolution)
X = smc.deconvolution::brain_heart_spec_ge
X = smc.deconvolution::hvg_selection(X)
M = smc.deconvolution::proportions
Y = smc.deconvolution::create_mixture_samples(X,M)
set.seed(10)
est_props = smc.deconvolution::smc_deconvolution(Y=Y,K=2,L=10)
M_est = smc.deconvolution::combine_est_props(est_props[[1]],M)
smc.deconvolution::simulation_summary(est_props[[2]])
```

Mean Run Time: 20.0908452 minutes (sd = 0.3825698 minutes)

```r
smc.deconvolution::scatter_plot(M,M_est)
```

Parallelization:

```r
Y = smc.deconvolution::create_mixture_samples(X,M)
set.seed(10)
est_props = smc.deconvolution::smc_deconvolution(Y=Y,K=2,L=10, parallel = T,
                                                 num_cores = 8)
```

```
M_est = smc.deconvolution::combine_est_props(est_props[[1]],M)
est_props[[2]]
```

Run Time: 1.703375 hours

$\frac{20.0908452*10-60*1.703375}{20.0908452*10} = 0.4912982$

Using parallelization reduces the run time of the 10 iterations by approximately 49%.

```
X = smc.deconvolution::brain_heart_spec_ge
X = smc.deconvolution::hvg_selection(X)
M = prop_rdirichlet(10,10,2,colnames(X))
Y = smc.deconvolution::create_mixture_samples(X,M)
set.seed(10)
est_props = smc.deconvolution::smc_deconvolution(Y=Y,K=2,L=10,N=40, parallel = T,
                                                 num_cores = 8)
M_est = smc.deconvolution::combine_est_props(est_props[[1]],M)
smc.deconvolution::simulation_summary(est_props[[2]])
smc.deconvolution::scatter_plot(M,M_est)
```

1.331589 hours

We have uploaded the SMC-Deconvolution package to GitHub, and it passes the travis-ci check:

https://github.com/kpcoleman/SMC-Deconvolution

Introduction

With technological advances in the field of genetics has come a vast increase in the amount of available gene expression (GE) data. These data have many applications, including in the study of the genetc basis of diseases. Though there is an abundance of GE data, downstream analysis may be inhibited by the lack of single-cell resolution in many of these datasets. Specifically, samples in GE datasets without single-cell resolution can contain multiple cells, potentially of different cell types. A lack of knowledge of the cell-type compositions of the samples in such datasets can cause limitations for analysis and application of these data. As a result, many cell-type deconvolution algorithmns have been developed to estimate the proportions of cells of given types across samples in heterogeneous GE datasets that lack single-cell resolution. One such method, developed by Ogundijo and Wang (2017), utilizes a sequential Monte Carlo (SMC) approach to estimate cell-type specific gene expression levels and cell-type proportions in heterogeneous GE datasets. The intention of this project was to replicate some of the simulations performed by these authors, as well as extend their investigation by considering multiple datasets with samples composed of different proportions of cell types.

Methods

Highly Variable Gene Selection

The SMC deconvolution algorithm is used to estimate the proportions of cell types across heterogeneous samples in a GE matrix. This method requires as input a GE matrix of heterogeneous samples, Y, as well as the total number of cell types, K, in the samples of Y. The algorithm then provides the estimated proportion of the K cell types in each of the samples of Y. In some cases, a matrix of cell-type specific gene expression profiles, X, may be available. For such situations, we included a function in our package to, prior to deconvolution, reduce the dimensionality of Y by only selecting counts for the top $(100 * p)\%$ highly variable genes (HVGs) in X, where p is set to 0.05 by default. HVGs are genes that show the most variation in expression levels across different cell types, and are therefore likely more informative for analyses investigating cell-type composition compared to genes that show less variability across cell types (Yip, Sham, and Wang 2019). HVG selection is commonly used for the clustering of cells in single-cell RNA-sequencing (scRNA-seq) data (Andrews and Hemberg 2018), and has been incorporated as a step prior to performing cell-type deconvolution (Elosua-Bayes et al. 2021). Selection of HVGs both greatly reduces computation time and has been shown to lead to accurate results when compared with including all genes for a given method.

Distributions of Model Parameters

After HVG selection is performed, we run the SMC deconvolution algorithm. This algorithm treats the proportions of each cell type in the samples in Y, $M = [\underline{m_1} \cdots \underline{x_J}]$ as unknown parameters to be estimated by the model. Here, $\underline{m_j} = (m_{1j} \cdots x_{Kj})^T$ are the proportions of cell-types $k = 1, ..., K$ that compose sample j in Y. The counts in the cell-type specific gene expression matrix $X = [\underline{x_1} \cdots \underline{x_K}]$, are also treated as unknown parameters, where $\underline{x_k} = (x_{1k} \cdots x_{Ik})^T$ contains the cell-type specific counts for genes $i = 1, ..., I$ for cell type k. There is also an unknown precision parameter, $\lambda = \mathrm{Var}^{-1}(y_{ij})$, resulting in a total of $(J * K + I * K + 1)$ model parameters. Let θ denote the set of all model parameters. Given θ, the distribution of the count of gene i in sample j of Y is:

$$p(y_{ij}|\theta) = \phi\Big(y_{ij}\Big|\sum_{k=1}^{K} x_{ik} m_{kj}, \lambda^{-1}\Big),$$

where $\phi(\cdot|\mu, \sigma^2)$ is the pdf of a $N(\mu, \sigma^2)$ distribution. Thus, the joint distribution of all gene counts in Y is:

$$\begin{aligned} p(Y|\theta) &= \prod_{i=1}^{I}\prod_{j=1}^{J} \phi\Big(y_{ij}\Big|\sum_{k=1}^{K} x_{ik} m_{kj}, \lambda^{-1}\Big) \\ &\propto \lambda^{IJ/2} \exp\Big\{-\frac{\lambda}{2}\sum_{i=1}^{I}\sum_{j=1}^{J}\Big(y_{ij} - \sum_{k=1}^{K} x_{ik} m_{kj}\Big)^2\Big\} \end{aligned}$$

The prior distribution for the cell-type k specific count of gene i, x_{ik} is given by:

$$\pi(x_{ik}) = \phi(x_{ik}|\mu_{ik}, \nu_{ik}^{-1}),$$

where μ_{ik} and ν_{ik}^{-1} are hyperparameters that are assumed known. By default, ν_{ik}^{-1} is set to 1000 and μ_{ik} is sampled from a $\mathrm{Unif}(0, 100)$ distribution, $i = 1, ..., I,\ k = 1, ..., K$. The prior distribution for the proportion of cell type k in sample j, m_{kj}, is:

$$\pi(m_{kj}) = \phi(m_{kj}|\mu_{kj}, \nu_{kj}^{-1}),$$

where μ_{kj} and ν_{kj}^{-1} are assumed known and set to 0 and 0.01, respectively. The prior distribution for λ is:

$$\pi(\lambda) \sim \text{Gamma}(\alpha, \beta),$$

where α and β are assumed known and set to 2 and 10000, respectively.

The SMC algorithm defines, at time t, the intermediate posterior distribution for the model parameters as:

$$p_t(\theta|Y) = \pi(\theta)p(Y|\theta)^{\epsilon_t},$$

where $N = 40$ samples are taken by default and

$$\{\epsilon_t\}_{t=1}^{T} = \{0, \epsilon_2, ..., \epsilon_{T-1}, 1\}.$$

For the algorithm we implemented,

$$\epsilon_t = \epsilon_{t-1} + 10^{-4},\ t = 2, ..., T \implies T = 1001.$$

.

As opposed to other Monte Carlo-based methods, defining the posterior distribution in this way gradually increases the influence of the distribution $p(Y|\theta)$. Then, for time t, a joint distribution for the model parameters at times $b = 1, ..., t$, $\theta_{1:t} = \{\theta_1, ..., \theta_t\}$, is defined by:

$$\tilde{p}_t(\theta_{1:t}) \propto \pi(\theta_t)p(Y|\theta_t)^{\epsilon_t} \prod_{b=1}^{t-1} \mathcal{L}_b(\theta_{b+1}, \theta_b),$$

where $\mathcal{L}_b(\theta_{b+1}, \theta_b)$ is a backwards Markov kernel. The goal of the SMC algorithm is to obtain N samples from $\tilde{p}_t(\theta_{1:t})$ for each $t = 1, ..., T$. However, the difficulty associated with sampling from this distribution leads to the construction of an importance distribution that is more suitable. At time t, this distribution is defined by:

$$q_t(\theta_{1:t}) = q_1(\theta_1) \prod_{f=2}^{t} \mathcal{K}_f(\theta_{f-1}, \theta_f),$$

where $\mathcal{K}_f(\theta_{f-1}, \theta_f)$ is a forwards Markov kernel.

A set of importance weights are calculated at each time t to account for sampling the parameters from the importance distribution q_t as opposed to the posterior distribution p_t. For sample n at time $t-1$, the unnormalized importance weight, $\tilde{w}_{t-1}^n$ is calculated as:

$$\tilde{w}_{t-1}^n \propto \frac{\tilde{p}_t(\theta_{1:t}^n)}{q_t(\theta_{1:t}^n)} = \frac{p_{t-1}(\theta_{t-1}^n|Y) \prod_{b=1}^{t-2} \mathcal{L}_b(\theta_{b+1}^n, \theta_b^n)}{q_1(\theta_1^n) \prod_{f=2}^{t-1} \mathcal{K}_f(\theta_{f-1}^n, \theta_f^n)}$$

These weights are then normalized to obtain the importance weights at time $(t-1)$:

$$w_{t-1}^n = \frac{\tilde{w}_{t-1}^n}{\sum_{n'=1}^{N} \tilde{w}_{t-1}^{n'}}.$$

The relationship between the unnormalized weights for sample n at times $t-1$ and t can be shown to be:

$$\tilde{w}_t^n \propto \tilde{w}_{t-1}^n \frac{p_t(\theta_t^n|Y)\mathcal{L}_{t-1}(\theta_t^n, \theta_{t-1}^n)}{p_{t-1}(\theta_{t-1}^n|Y)\mathcal{K}_t(\theta_{t-1}^n, \theta_t^n)}.$$

Each time the normalized weights are computed, we calculate the effective sample size (ESS) by:

$$ESS = \frac{1}{\sum_{n=1}^N (w_t^n)^2}.$$

A small ESS implies that some of the samples have a relatively large weight compared to others. To remedy this, a resampling is performed at any time t such that $ESS < \frac{N}{10}$. Specifically, N samples are drawn with replacement according to the weights $w_t^1, ..., w_t^N$ and the weights are set to $1/N$. In order sample the unknown parameters from $q_t(\theta_{1:t})$, the posterior distribution for the parameters at time t, $p_t(\theta|Y)$ is required. Samples from $p_t(\theta|Y)$ are obtained using a Gibbs sampler, i.e. by sampling from the conditional posterior distributions of the model parameters (Gelman et al. 2013). Specifically, at time t, the posterior distribution of m_{kj} conditioned on the other parameters, is given by:

$$p_t(m_{kj}|\cdot) = \phi\left\{m_{kj}\middle|\frac{\mu_{kj}\nu_{kj} + \epsilon_t\lambda\sum_{i=1}^I (y_{ij}x_{ik} - (\sum_{k'\neq k} x_{ik'}m_{k'j})x_{ik})}{\nu_{kj} + \epsilon_t\lambda\sum_{i=1}^I x_{ik}^2}, \frac{1}{\nu_{kj} + \epsilon_t\lambda\sum_{i=1}^I x_{ik}^2}\right\},$$

where "$\cdot$" indicates Y and the most recent updates of the other parameters. The conditional posterior distribution of x_{ik} at time t is given by:

$$p_t(x_{ik}|\cdot) = \phi\left\{x_{ik}\middle|\frac{\mu_{ik}\nu_{ik} + \epsilon_t\lambda\sum_{j=1}^J (y_{ij}m_{kj} - (\sum_{k'\neq k} x_{ik'}m_{k'j})m_{kj})}{\nu_{ik} + \epsilon_t\lambda\sum_{j=1}^J m_{kj}^2}, \frac{1}{\nu_{ik} + \epsilon_t\lambda\sum_{j=1}^J m_{kj}^2}\right\},$$

and λ is sampled according the conditional posterior distribution:

$$p_t(\lambda|\cdot) \sim \text{Gamma}\left\{\alpha + \frac{\epsilon_t IJ}{2}, \beta + \frac{\epsilon_t}{2}\sum_{i=1}^I\sum_{j=1}^J\left(y_{ij} - \sum_{k=1}^K x_{ik}m_{kj}\right)^2\right\}.$$

In addition to the posterior distribution, the Markov kernels $\mathcal{K}_t$ and $\mathcal{L}_{t-1}$ need to be specified. Following Nguyen *et al.* (2016), the kernel $\mathcal{K}_t$ is chosen to be:

$$\mathcal{K}_t(\theta_{t-1}, \theta_t) = p_t(\theta_t),$$

and the kernel $\mathcal{L}_{t-1}$ is:

$$\mathcal{L}_{t-1}(\theta_t, \theta_{t-1}) = \frac{p_t(\theta_{t-1})\mathcal{K}_t(\theta_t, \theta_{t-1})}{p_t(\theta_t)}.$$

Defining the forward and backward Markov kernels in this way results in the following form of the unnormalized weight at time t for sample n:

$$\tilde{w}_t^n \propto \tilde{w}_{t-1}^n p(Y|\theta_{t-1}^n)^{\epsilon_t - \epsilon_{t-1}}$$

Steps of SMC algorithm

For time $t = 1$ of the SMC algorithm, N samples for the unknown parameters λ, $\{m_{kj}\}$, and $\{x_{ik}\}$ from their respective prior distributions defined above, and the weights $w_1^1, ..., w_1^N$ are set to $\frac{1}{N}$. Then, for times $t = 2, ..., T$, the following steps are peformed:

(1) The unnormalized weights $\tilde{w}_t^n$ are calculated as:

$$\tilde{w}_t^n = w_{t-1}^n p(Y|\theta_{t-1}^n)^{\epsilon_t - \epsilon_{t-1}},$$

and then normalized to sum to 1.

(2) The ESS is calculated and resampling occurs if necessary.

(3) Sample m_{kj} N times from $p_t(m_{kj}|\cdot)$, $k = 1, ..., K$, $j = 1, ..., J$.

(4) Sample x_{ik} N times from $p_t(x_{ik}|\cdot)$, $i = 1, ..., I$, $k = 1, ..., K$.

After T iterations, the estimated cell-type proportions across the samples in Y are calculated as:

$$\hat{m}_{kj} = \sum_{n=1}^{N} w_T^n (m_{kj}^n)_T, \qquad k = 1, ..., K, j = 1, ..., J.$$

Results

We utilized the SMC deconvolution method to analyze the

The SMC algorithm is unsupervised, and so does not rely on an annotated reference dataset. As a result, this method does not provide the name of the cell type associated with a given proportion estimate.

$\theta = \{\lambda, X, M\}x_{ik}, m_{kj}\}$ are the parameters estimated by the model

$$\lambda \sim \text{Gamma}(\alpha, \beta)$$

$$w_1^n = 1/N$$

$$\tilde{w}_t^n = w_{t-1}^n p(Y|\theta_{t-1})^{\epsilon_t - \epsilon_{t-1}}$$

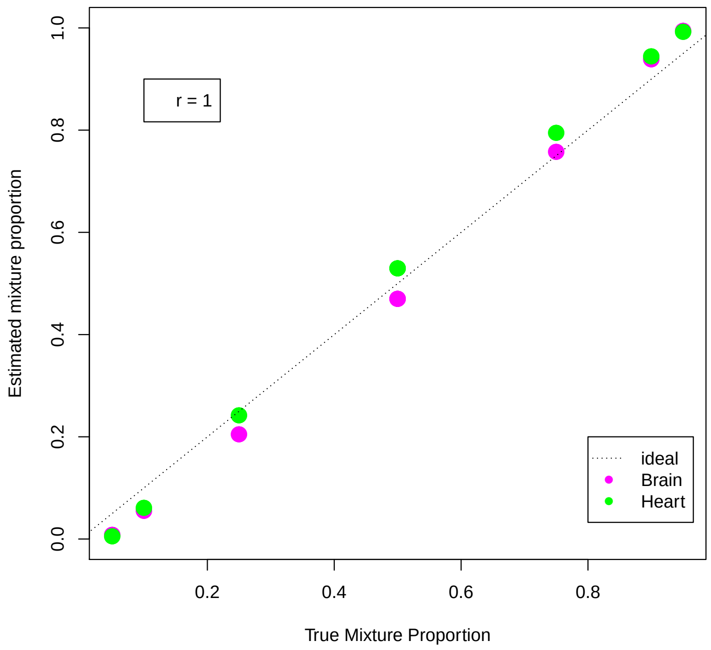

Figure 1: Scatter plot of true versus estimated cell-type proportions (Proportions given in paper)

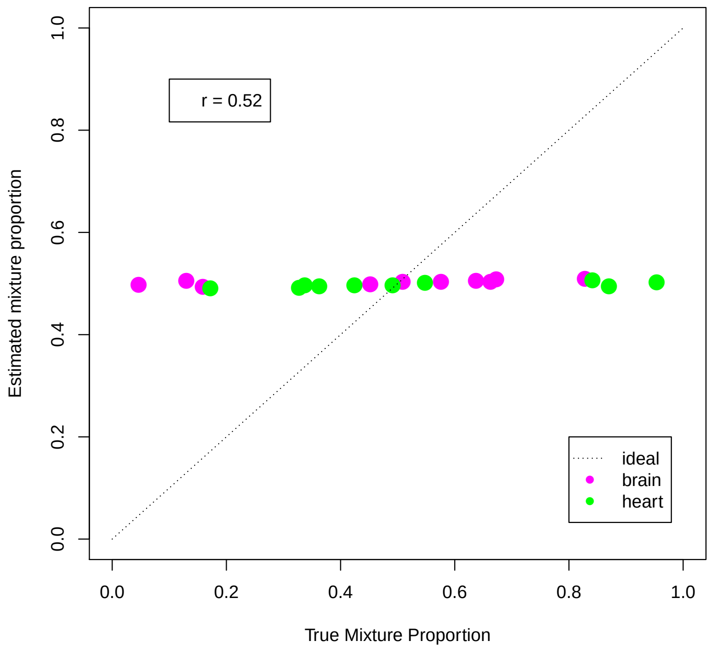

Figure 2: Scatter plot of true versus estimated cell-type proportions (Proportions generated from Dirichlet distribution for 10 samples)

Bibliography

Andrews, T S, and M Hemberg. 2018. “Identifying Cell Populations with scRNASeq.” *Molecular Aspects of Medicine* 59: 114–22.

Elosua-Bayes, M, P Nieto, E Mereu, I Gut, and H Heyn. 2021. “SPOTlight: Seeded Nmf Regression to Deconvolute Spatial Transcriptomics Spots with Single-Cell Transcriptomes.” *Nucleic Acids Research.*

Gelman, A, J Carlin, H Stern, D B Dunson, A Vehtari, and D B Rubin. 2013. *Bayesian Data Analysis, Third Edition.* London: Chapman & Hall.

Nguyen, T LT, F Septier, G W Peters, and Y Delignon. 2016. “Efficient Sequential Monte-Carlo Samplers for Bayesian Inference.” *IEEE Transactions on Signal Processing* 64: 1305–19.

Ogundijo, O E, and X Wang. 2017. “A Sequential Monte Carlo Approach to Gene Expression Deconvolution.” *PLoS ONE* 12: 10.

Yip, S H, P C Sham, and J Wang. 2019. “Evaluation of Tools for Highly Variable Gene Discovery from Single-Cell Rna-Seq Data.” *Briefings in Bioinformatics* 20: 1583–9.