A Sequential Monte Carlo Approach to Gene Expression Deconvolution

Emmanuelle Dankwa Natalia Garcia Martin William Thomas Yuxi Jiang

October 18, 2019

Abstract

DNA microarrays have become one of the main methods of obtaining gene expression data and are an important resource for identifying genetic variation. Such data can be obtained from either pure or heterogeneous biological samples. In the heterogeneous case, the measured expression can be attributed to multiple cell types. The presence of heterogeneity can make the analysis of such data more complicated, and while methods exist to physically separate cell types in biological samples, this can be time consuming and can potentially lead to contamination of samples. Within this report, we examine a sequential Monte Carlo approach to parameter estimation, treating the cell type proportions and the cell specific expressions as the parameters of interest. This approximates the posterior distribution using a sample of weighted observations, obtained using a sequence of artificial target distributions. This algorithm is then evaluated on real datasets with two or three cell types, before being compared to a nonnegative matrix factorization (NMF) approach.

1 Introduction

Using high-throughput gene expression data from DNA microarrays, we have the ability to identify sources of genetic variation. With this information, we have the potential to improve medical diagnoses, treatment prescription and drug design (Ogundijo and Wang, 2017). However, heterogeneous biological samples containing more than one cell type can complicate the analysis of such data. This is because each cell type within the sample may make a different contribution to the measured gene expression. Methods such as laser microdissection and flow cytometry are able to physically separate cell types in biological samples, but this can be expensive, time consuming and result in sample contamination. Alternatively, several computational methods have been proposed, such as non-negative matrix factorization (NMF) algorithms (Repsilber et al., 2010) and probabilistic approaches based on Markov chain Monte Carlo (MCMC) methods (Erkkilä et al., 2010). However, these methods generally require some prior biological knowledge. Within this report, we will examine a sequential Monte Carlo (SMC) approach (Ogundijo and Wang, 2017) to estimating the cell type proportions within biological samples without the use of any prior information.

The SMC approach used by Ogundijo and Wang (2017) builds upon a Bayesian framework where the cell type proportions are taken to be the unknown model parameters. The overall aim of the algorithm is to efficiently approximate the joint posterior of all unknown model parameters using a set of weighted samples, operating in a similar manner to sequential importance sampling. By using SMC algorithms, we avoid some of the shortcomings of MCMC methods and have the option to parallelise the algorithm to reduce computational time.

In Section 2, we present the general theory of sequential Monte Carlo methods, demonstrating how adapting an importance sampler can lead to an SMC algorithm. In Section 3, we then present an SMC algorithm for gene expression deconvolution, as proposed by Ogundijo and Wang (2017), as well as the associated theory and justification for the various choices and steps. We then apply the SMC algorithm to a real, Affymetrix dataset consisting of gene expression from two cell types in Section 4, before comparing our results to those obtained using an NMF algorithm. Finally, we discuss the limitations of the proposed SMC algorithm in Section 5, also highlighting some possible improvements and extensions.

2 General Sequential Monte Carlo

Consider a state-space model, where the unobserved hidden states, $\{x_t : t \in \mathbb{N} \cup \{0\}\}$, can be modelled as a Markov process with initial distribution $p(x_0)$ and transition probability $p(x_t|x_{t-1})$; and the observations $\{y_t : t \in \mathbb{N}\}$, with marginal distribution $p(y_t|x_t)$, are conditionally independent given the hidden process $\{x_t : t \in \mathbb{N}\}$.

To estimate the posterior distribution $p(x_{0:t}|y_{1:t})$ of the hidden states given the observations, one method would be to use the importance sampling algorithm, with importance weight given by

$$w_t = \frac{p(x_{0:t}|y_{1:t})}{\pi(x_{0:t}|y_{1:t})} = \frac{p(y_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|y_{1:t-1})}{\pi(x_{0:t}|y_{1:t})},$$

where $\pi(\cdot|\cdot)$ denotes the proposed sampling distribution. Each time a new observation y_{t+1} is available, the importance weight needs to be recalculated using all of the observations $\{y_{1:t+1}\}$ at each iteration to obtain one sample. The computational complexity increases with time, which means the importance sampling algorithm is not adequate for recursive estimation under this construction.

2.1 Sequential Importance Sampling

To modify the importance sampling algorithm so that it is suitable for recursive estimation, we need to reconstruct another proposal sampling distribution so that the density function at time $t-1$ is the marginal distribution of the density function at time t, then iteratively, we have

$$\begin{aligned}\pi(x_{0:t}|y_{1:t}) &= \pi(x_t|x_{0:t-1}, y_{1:t})\pi(x_{0:t-1}|y_{1:t-1}) \\ &= \pi(x_0)\prod_{k=1}^{t}\pi(x_t|x_{0:k-1}, y_{1:k}).\end{aligned}$$

By the construction of proposal distribution, the importance weight can be updated sequentially in time by

$$\begin{aligned}w_t &\propto \frac{p(y_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|y_{1:t-1})p(x_{t-1}|x_{t-2})p(x_{0:t-2}|y_{1:t-2})}{\pi(x_t|x_{0:t-1}, y_{1:t})\pi(x_{0:t-1}|y_{1:t-1})} \\ &\propto w_{t-1}\frac{p(y_t|x_t)p(x_t|x_{t-1})}{\pi(x_t|x_{0:t-1}, y_{1:t})},\end{aligned}$$

which is the importance weight used for sequential importance sampling (SIS). Unfortunately, SIS suffers from particle degeneracy, where the skewness of the distribution of the importance weights increase as t increases (Doucet et al., 2001). In particular, some particles can have very small weights while some particles can have very large weights. One possible remedy would be to increase the effect of particles with large weights and pay less attention to particles with negligible weights. This is the main idea behind sequential importance sampling with resampling (SISR).

2.2 Sequential Importance Sampling with Resampling

By incorporating resampling into an SIS algorithm, we are able to increase the chance of propagating 'weighty' particles, while ensuring that 'lighter' particles are less likely to be propagated (Picchini, 2016). We outline below the resampling procedure employed by Ogundijo and Wang (2017):

1. The weight of each particle, w_t^n, is considered as the probability of drawing such a particle from the set of all particles at time t.

2. N particles are randomly drawn with replacement from the set of particles at time t. The consideration in point 1 ensures that there is a higher probability of obtaining heavier particles in the set of selected particles.

3. Replace the old set of particles with the set of newly selected particles. Note that both sets have the same number of elements, N. Set the weights of all elements in this new set to $1/N$.

The bootstrap filter is the simplest example of a sequential importance sampling with resampling (SISR) algorithm (Picchini, 2016) and can be seen in Algorithm 1. This is also the most basic form of a sequential Monte Carlo algorithm. In the following section, we will describe how we can extend the SMC framework to perform gene expression deconvolution, and present an SMC algorithm proposed by Ogundijo and Wang (2017).

Algorithm 1 Bootstrap filter

1. Initialisation: $t = 1$

   for $n = 1$ to N **do**
   draw a sample $x_0^n \sim p(x_0)$
   assign weights $\tilde{w}_0^n = 1/N,\ n = 1, \ldots, N$
   end for
   Set $t = 2$

2. Importance sampling step

   for $n = 1$ to N **do**
   draw a sample $\tilde{x}_t^n \sim p(x_t | x_{t-1}^n)$ and set $\tilde{x}_{0:t}^n = (\tilde{x}_{0:t-1}^n, \tilde{x}_t^n)$
   end for
   for $n = 1$ to N **do**
   - Compute the unnormalised importance weights: $\tilde{w}_t^n = p(y_t | \tilde{x}_t^n)$.
   - Normalise the weights: $w_t^n = \frac{\tilde{w}_t^n}{\sum_{l=1}^N \tilde{w}_t^l},\ n = 1, \ldots, N.$

   end for

3. Selection step

   - Resample with replacement N particles $(x_{0:t}^n :\ n = 1, \ldots, N)$ from the set $(\tilde{x}_{0:t}^n :\ n = 1, \ldots, N)$ according to the importance weights.
   - Set $t \leftarrow t + 1$ and if $t < T$ go to step 2.

3 Sequential Monte Carlo for gene expression deconvolution

Let $\boldsymbol{Y}$ denote the I x J heterogeneous gene expression matrix, where I indicates the number of genes and J the number of samples. We assume that the number of cell types K is known and that each sample has the same number of cell types present, but in varying proportions. The relationship between the expression value of pure and mixed samples is assumed linear and the expression level of gene i in sample j is given by the sum of its expression across the K cell types:

$$y_{ij} = \sum_{k=1}^{K} x_{ik} m_{kj} + e_{ij}, \quad i = 1, \ldots, I,\ j = 1, \ldots, J,$$

where x_{ik} denotes the expression of gene i in cell type k, m_{kj} denotes the proportion of cell type k in sample j and e_{ij} is additive Gaussian noise with mean zeros and precision λ (that is, variance $1/\lambda$). This can be written in matrix form as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{M} + \boldsymbol{E},$$

where the I x K expression level matrix across cell types $\boldsymbol{X}$ and the K x J cell proportions matrix $\boldsymbol{M}$ are both unknown. Note that $\boldsymbol{M}$ is non-negative with columns that sum to 1. Our goal is to infer $\boldsymbol{X}$, $\boldsymbol{M}$ and λ ogeneous gene expression matrix $\boldsymbol{Y}$.

The data point y_{ij} corresponds to the sum of the cell-type specific expressions of gene i weighted by their proportions in sample j plus the Gaussian distributed noise e_{ij}. Therefore,

$$p(y_{ij} | x_{i,:}, m_{:,j}, \lambda) = \mathcal{N}(\boldsymbol{x_{i:}}, \boldsymbol{m_{:j}}, \lambda^{-1}) = \mathcal{N}\left(\sum_{k=1}^{K} x_{ik} m_{kj}, \lambda^{-1}\right).$$

Then, assuming independent and identically distributed measurements, we can write the joint likelihood function as

$$p(\boldsymbol{Y}|\boldsymbol{\theta}) = \prod_{i=1}^{I}\prod_{j=1}^{J} p(y_{ij}|\boldsymbol{x}_{i,:}, \boldsymbol{m}_{:,j}, \lambda),$$

where $\boldsymbol{\theta} = \{\lambda,\ x_{ik},\ m_{kj} :\ i = 1, \ldots, I,\ j = 1, \ldots, J,\ k = 1, \ldots, K\}$ is the vector of unknown parameters.

We will define a data generating model, impose prior distributions on all the unknown model parameters, derive a sequence of target distributions for such parameters and present an SMC algorithm for efficiently estimating these.

3.1 Densities of model parameters

In this section, we present the prior densities chosen by Ogundijo and Wang (2017) for each of the model parameters introduced in the previous section. We also present the derivation of the sequence of target distributions for the cell-type specific expressions.

3.1.1 Cell-type specific expressions

For the prior distribution of x_{ik}, the authors choose a Gaussian distribution with known mean μ_{ik} and variance ν_{ik}^{-1}, where ν_{ik} is the precision. The authors explain that the choice of prior distribution is motivated by the property of conjugate priors, which ensure that the sequence of target distributions are also Gaussian, given that the likelihood is Gaussian (Gelman et al., 2013).

To obtain the sequence of target distributions, $\pi_t(x_{ik}|\cdot)$, for the cell-type specific expressions, let $p(x_{ik}|\mu_{ik}, \nu_{ik})$ and $\prod_{j=1}^{J} p(y_{ij}|\sum_{k'=1}^{K} x_{ik'}m_{k'j}, \lambda)$, $i = 1, \ldots, I;\ k = 1, \ldots, K;\ t = 1, \ldots, T$; represent the prior distribution and likelihood function respectively of the cell-type specific expression. Then,

$$\begin{aligned}\pi_t(x_{ik}|\cdot) &\propto p(x_{ik}|\mu_{ik}, \nu_{ik}) \left[\prod_{j=1}^{J} p(y_{ij}|\sum_{k'=1}^{K} x_{ik'}m_{k'j}, \lambda)\right]^{\epsilon_t} \\ &\propto \exp\left\{-\frac{A_{ik}^t}{2}\left[x_{ik} - \frac{B_{ik}^t}{A_{ik}^t}\right]^2\right\}.\end{aligned}$$

Thus, $\pi_t(x_{ik}|\cdot) \sim \mathcal{N}\Big(\frac{B_{ik}^t}{A_{ik}^t}, \frac{1}{A_{ik}^t}\Big)$, where $\mathcal{Y}_{ijk} = \sum_{k'\neq k} x_{ik'}m_{k'j}$, $A_{ik}^t = \nu_{ik} + \epsilon_t\lambda\sum_{j=1}^{J} m_{kj}^2$ and $B_{ik}^t = \mu_{ik}\nu_{ik} + \epsilon_t\lambda\left(\sum_{j=1}^{J} y_{ij}m_{kj} - \sum_{j=1}^{J}\mathcal{Y}_{ijk}m_{kj}\right)$. A full derivation of this relationship is given in the Appendix.

3.1.2 Cell type proportions

Again, utilising the property of conjugate priors, the authors choose a Gaussian prior for m_{kj} with known mean μ_{kj} and known variance ν_{kj}^{-1}. Following a similar derivation method as in section 3.1.1, the authors derive the sequence of target distributions $\pi_t(m_{kj}|\cdot)$ for the cell type proportions:

$$\pi_t(m_{kj}|\cdot) \sim \mathcal{N}\Big(\frac{V_{kj}^t}{U_{kj}^t}, \frac{1}{U_{kj}^t}\Big),$$

$$\text{where}\ \ U_{kj}^t = \nu_{kj} + \epsilon_t\lambda\sum_{i=1}^{I} x_{ik}^2\ \ \text{and}\ \ V_{kj}^t = \mu_{kj}\nu_{kj} + \epsilon_t\lambda\Big(\sum_{i=1}^{I} y_{ij}x_{ik} - \sum_{i=1}^{I}\mathcal{Y}_{ijk}x_{ik}\Big).$$

3.1.3 Noise precision

A Gamma prior with parameters α and β is chosen for the precision parameter to ensure that the sequence of target distributions, $\pi_t(\lambda|\cdot)$, follows a Gamma distribution, given that the likelihood is Gaussian (Gelman et al., 2013). The authors show that $\pi_t(\lambda|\cdot) \sim \text{Gamma}(\hat{\alpha}, \hat{\beta})$, where $\hat{\alpha} = \alpha + \frac{\epsilon_t IJ}{2}$ and $\hat{\beta} = \beta + \frac{\epsilon_t}{2}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(y_{ij} - \sum_{k=1}^{K} x_{ik}m_{kj}\right)^2$.

3.2 Estimation procedure

In this section, we give the theoretical background of the estimation procedure employed by Ogundijo and Wang (2017) for the gene expression deconvolution problem. As explained earlier in the introduction, sampling from the target distribution $p(\boldsymbol{\theta}|\boldsymbol{Y})$ may be prohibitive thus warranting the need for an approximate procedure for accomplishing this task. To circumvent this difficulty, SMC samplers consider a sequence of intermediate target distributions, $\{\pi_t\}_{t=1}^T$, such that the prior, $p(\theta)$, is equal to π_1 and $p(\theta|\boldsymbol{Y}) = \pi_T$ (Nguyen et al., 2016). The sequence $\{\pi_t\}_{t=1}^T$ is generally chosen to have the form:

$$\pi_t(\boldsymbol{\theta}) = \frac{\boldsymbol{\Psi}_t(\boldsymbol{\theta})}{\boldsymbol{Z_t}} \propto p(\boldsymbol{\theta})p(\boldsymbol{Y}|\boldsymbol{\theta})^{\epsilon_t},$$

where $\boldsymbol{Z_t} = \int_\Theta p(\boldsymbol{\theta})p(\boldsymbol{Y}|\boldsymbol{\theta})^{\epsilon_t} d\boldsymbol{\theta}$ and $\{\epsilon_t\}_{t=1}^T$ is defined as a non-decreasing temperature schedule with $\epsilon_1 = 0$ and $\epsilon_T = 1$ (Ogundijo and Wang, 2017; Nguyen et al., 2016). As ϵ_t is increased, we gradually introduce the effect of the likelihood, so that our final target density corresponds to the desired posterior distribution.

Treating the estimation as a Bayesian filtering problem, the authors introduce the sequence $\{\tilde{\pi}\}_{t=1}^T$ of joint target distributions up until t, which admits π_t as marginals. This is given by:

$$\tilde{\pi}_t(\boldsymbol{\theta}_{1:t}) = \frac{\tilde{\Psi}_t(\boldsymbol{\theta}_{1:t})}{\boldsymbol{Z_t}},$$

where $\tilde{\Psi}_t(\boldsymbol{\theta}_{1:t}) = \Psi_t(\boldsymbol{\theta}_t)\prod \mathcal{L}_b(\boldsymbol{\theta}_{b+1}, \boldsymbol{\theta}_b)$ and $\{\mathcal{L}_d\}_{d=1}^{t-1}$ are the backward transition kernels representing the probability density of moving from θ_{t+1} to θ_t.

Sampling from $\tilde{\pi}_t(\boldsymbol{\theta}_{1:t})$ is often not straightforward and there is thus the need to employ an approximate distribution, the importance distribution, from which sampling is relatively easier. For this problem, Ogundijo and Wang (2017) define the importance distribution at time t, as

$$q_t(\boldsymbol{\theta}_{1:t}) = q_1(\boldsymbol{\theta}_1)\prod_{f=2}^{t} \mathcal{K}_f(\boldsymbol{\theta}_{f-1}, \boldsymbol{\theta}_f) \tag{1}$$

where $\mathcal{K}_t(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t)$ represents the probability of moving from $\boldsymbol{\theta}_{t-1}$ to $\boldsymbol{\theta}_t$, i.e. $\{\mathcal{K}_d\}_{d=2}^t$ are forward Markov transition kernels.

To account for the fact that samples are drawn from an approximate distribution rather than the true distribution, importance weights are computed, at time $t-1$, for all N samples drawn from the importance distribution and these weights are normalized accordingly. The (unnormalized) importance weights are computed using the expression below:

$$\tilde{w}_{t-1}^n \propto \frac{\tilde{\pi}_{t-1}(\boldsymbol{\theta}_{1:t-1}^n)}{q_{t-1}(\boldsymbol{\theta}_{1:t-1}^n)} = \frac{\pi_{t-1}(\boldsymbol{\theta}_{t-1}^n)\prod_{d=1}^{t-2}\mathcal{L}_d(\boldsymbol{\theta}_{d+1}^n, \boldsymbol{\theta}_d^n)}{q_1(\boldsymbol{\theta}_1^n)\prod_{r=2}^{t-1}\mathcal{K}_r(\boldsymbol{\theta}_{r-1}^n, \boldsymbol{\theta}_r^n)}.\ n = 1, \ldots, N \tag{2}$$

Denoting the normalized weights by w_{t-1}^n, we have that $\{\boldsymbol{\theta}_{1:t-1}^n, w_{t-1}^n\}_{n-1}^N$ approximate the distribution $\tilde{\pi}_{t-1}$, i.e. at time, $t-1$. The forward transition kernel is employed in the propagation of particles from $\tilde{\pi}_{t-1}$ to $\tilde{\pi}_t$ to obtain an approximation at time, t. Again, importance weights, $\tilde{w}_t^n$, are computed at time t, for each of these propagated particles:

$$\tilde{w}_t^n \propto \frac{\tilde{\pi}_t(\boldsymbol{\theta}_{1:t}^n)}{q_t(\boldsymbol{\theta}_{1:t}^n)}.$$

Ogundijo and Wang (2017) also derive the following expression for the relationship between $\tilde{w}_t^n$ and $\tilde{w}_{t-1}^n$:

$$\tilde{w}_t^n = \tilde{w}_{t-1}^n W_t(\boldsymbol{\theta}_{t-1}^n, \boldsymbol{\theta}_t^n); \qquad W_t(\boldsymbol{\theta}_{t-1}^n, \boldsymbol{\theta}_t^n) = \frac{\Psi_t(\boldsymbol{\theta}_t^n)\mathcal{L}_{t-1}(\boldsymbol{\theta}_t^n, \boldsymbol{\theta}_{t-1}^n)}{\Psi_{t-1}(\boldsymbol{\theta}_{t-1}^n)\mathcal{K}_t(\boldsymbol{\theta}_{t-1}^n, \boldsymbol{\theta}_t^n)}, \tag{3}$$

where $\{W_t(\boldsymbol{\theta}_{t-1}^n, \boldsymbol{\theta}_t^n)\}_{n=1}^N$ are the unnormalized incremental weights.

Algorithm 2 SMC sampler for gene expression deconvolution (Ogundijo and Wang, 2017)

1. Input the heterogeneous gene expression matrix $\boldsymbol{Y}$, α, β, $\{\mu_{kj},\ \nu_{kj},\ k = 1,\ldots,K,\ j = 1,\ldots,J\}$, $\{\mu_{ik},\ \nu_{ik},\ i = 1,\ldots,I,\ k = 1,\ldots,K\}$ and the temperature schedule $0 = \epsilon_1 < \epsilon_2 \cdots < \epsilon_T = 1$.
2. Set $t = 1$

   for $n = 1$ to N **do**
     draw a sample from `Gamma` $(\alpha,\ \beta)$
     for $k = 1$ to K **do**
       for $j = 1$ to J **do**
         draw a sample from $\mathcal{N}(\mu_{kj},\ \nu_{kj}^{-1})$
       end for
     end for
     for $i = 1$ to I **do**
       for $k = 1$ to K **do**
         draw a sample from $\mathcal{N}(\mu_{ik},\ \nu_{ik}^{-1})$
       end for
     end for
   end for
   Set $w_1^n = 1/N,\ n = 1,\ldots,N$.
3. **for** $t = 2$ to T **do**

   (i) Compute the unnormalised weights: $\tilde{w}_t^n = w_{t-1}^n \boldsymbol{p}(\boldsymbol{Y}|\boldsymbol{\theta}_{t-1})^{(\epsilon_t - \epsilon_{t-1})},\ n = 1,\ldots,N$.

   (ii) Normalise the weights: $w_t^n = \frac{\tilde{w}_t^n}{\sum_{l=1}^N \tilde{w}_t^l},\ n = 1,\ldots,N$.

   (iii) Compute ESS $= 1/\sum_{n=1}^N (w_t^n)^2$ and resample if ESS $< N/10$.

   (iv) Propagate the particles:

   for $n = 1$ to N **do**
     draw a sample from $\pi_t(\lambda|\cdot)$
     for $k = 1$ to K **do**
       for $j = 1$ to J **do**
         draw a sample from $\pi_t(m_{kj}|\cdot)$
       end for
     end for
     for $i = 1$ to I **do**
       for $k = 1$ to K **do**
         draw a sample from $\pi_t(x_{ik}|\cdot)$
       end for
     end for
   end for

   end for
4. Compute the parameter estimates as $\hat{\boldsymbol{\theta}} = \sum_{n=1}^N w_T^n \boldsymbol{\theta}_T^n$ and obtain $\hat{\boldsymbol{M}},\ \hat{\boldsymbol{X}}$ and $\hat{\lambda}$ from $\hat{\boldsymbol{\theta}}$.

3.2.1 Choice of kernel

It is worth noting that the performance of SMC algorithms highly depend on the choice of transition kernels (Nguyen et al., 2016). In their paper, Ogundijo and Wang (2017) draw on the results from previous work by Nguyen et al. (2016) which shows that in order to minimize the variance of the importance weights, the forward kernel, $\mathcal{K}_t(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t)$, should equal $\pi_t(\boldsymbol{\theta}_t)$. Also, Nguyen et al. (2016) present the following expression as a good approximation to the optimal backward kernel which minimizes

the variance of the importance weights, provided the difference between π_t and π_{t-1} is small:

$$\mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}) = \frac{\pi_t(\boldsymbol{\theta}_{t-1})\mathcal{K}_t(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1})}{\pi_t(\boldsymbol{\theta}_t)}. \tag{4}$$

3.3 The SMC algorithm for gene expression deconvolution

The full algorithm for the SMC sampler for gene expression deconvolution is presented as Algorithm 2. An additional step included in this algorithm is the introduction of a criterion for resampling particles. Ogundijo and Wang (2017) perform resampling only for steps where the effective sample size (ESS) is significantly less than the sample size, setting the significant threshold to $N/10$. The authors compute the ESS using the inverse of the sum of the squared weights of each particle in the current sample:

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

4 Results

4.1 SMC algorithm: two cell types example

In this section, we reproduced the experiment by Ogundijo and Wang (2017) using the Affymetrix dataset (Affymetrix, 2009), which contains heterogeneous expressions from human brain and heart cells. There are 33 samples in the dataset with the true mixing proportions given in Table 1. Before the dataset is analysed with the proposed SMC algorithm, the data was preprocessed using the robust multi-array average (RMA) procedure (Irizarry et al., 2003) involving steps of background adjustments, normalisation and summarisation. Only the heterogeneous samples S4 to S30 are used to estimate the matrix of cell proportions $\boldsymbol{M}$.

	S1-S3	S4-S6	S7-S9	S10-S12	S13-S21	S22-S24	S25-S27	S28-S30	S31-S33
Brain	0.00	0.05	0.10	0.25	0.50	0.75	0.90	0.95	1.00
Heart	1.00	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.00

Table 1: True cell type proportions for each sample in the Affymetrix dataset.

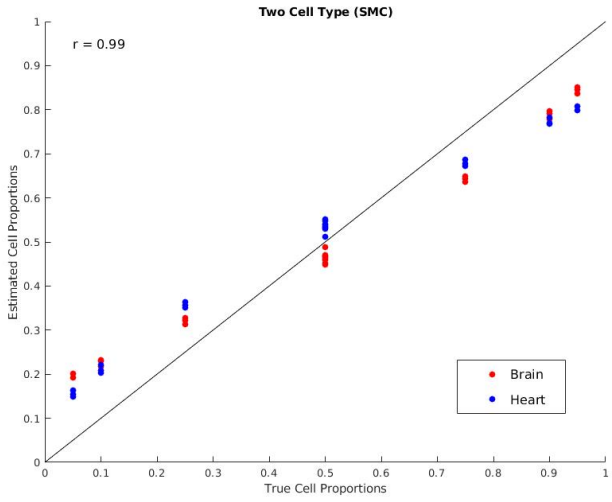

Figure 1: Plot of estimated versus true mixture proportions for the two cell types dataset using the proposed SMC method. The Pearson correlation coefficient for the estimated versus true proportions matrix is given in the upper left corner.

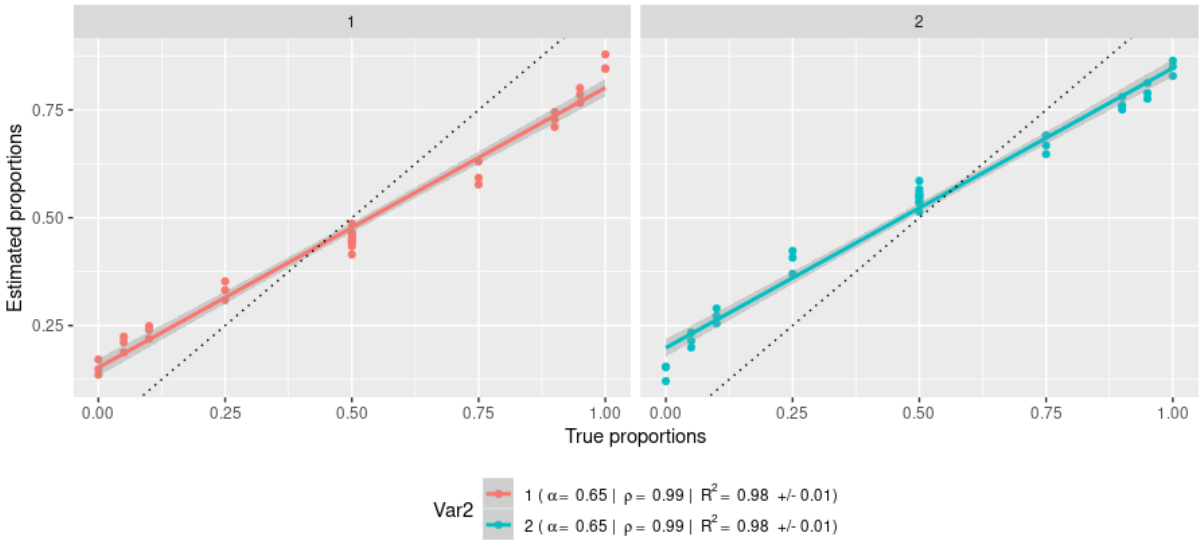

Figure 2: Plot of estimated versus true mixture proportions for the two cell types dataset using the *deconf* method. (Left) Brain cells. (Right) Heart cells. The algorithm converged after 8 iterations with an elapsed time of 0.998 seconds, giving a squared Pearson correlation coefficient of 0.98.

The proposed SMC algorithm was run with a smaller dataset that only includes 2000 randomly selected genes from the Affymetrix dataset, with $T = 5000$ and $N = 40$. The algorithm took 3.36 hours on 3.1 Ghz Intel 7 processors and the results are illustrated in Figure 1.

4.2 Comparison to other methods

The CellMix R package (Gaujoux and Seoighe, 2013a) contains several methods and utilities for performing both partial and complete gene expression deconvolution. Partial deconvolution methods assume that either signatures or proportions are available and use them to infer the unknown proportions and signatures respectively. Complete deconvolution methods infer both cell-type signatures and proportions from the global gene expression data, possibly using extra data such as marker genes to guide or seed the estimation. We will focus on the latter set of methods.

Nonnegative Matrix Factorization (Lee and Seung, 1999; Paatero and Tapper, 1994) is a popular choice since gene expression deconvolution can naturally be expressed as a matrix decomposition problem. The method *deconf* uses an alternate least-squares algorithm to estimate both cell proportions and cell-specific signatures from global expression data, as proposed by Repsilber et al. (2010). It is implemented as an NMF (Non-Negative Matrix Factorization) algorithm using the NMF package (Gaujoux and Seoighe, 2010). A new improved implementation is based on the Fast Combinatorial Nonnegative Least-Squares algorithm from Van Benthem and Keenan (2004) and Kim and Park (2007), as provided by the function *fcnnls* in the NMF package. This involves fitting in a completely unsupervised manner. It enables us to achieve great performance speed-up, being much faster than the original implementation. Unfortunately, the DSection method, which uses a Monte-Carlo-Markov-Chain (MCMC) approach to model the uncertainty in the proportion measurements, is no longer accessible.

	S1 - S3	S4- S6	S7- S9	S10-S12	S13-S15	S16-S18	S19-S21
Liver	1.00	0.00	0.00	0.05	0.70	0.25	0.70
Brain	0.00	1.00	0.00	0.25	0.05	0.70	0.25
Lung	0.00	0.00	1.00	0.70	0.25	0.05	0.05
	S22-24	S25-27	S28-30	S31-S33	S34-S36	S37-S39	S40-S42
Liver	0.45	0.55	0.50	0.55	0.50	0.60	0.65
Brain	0.45	0.30	0.30	0.30	0.40	0.35	0.34
Lung	0.10	0.25	0.20	0.15	0.10	0.05	0.01

Table 2: True cell type proportions for each sample in the GSE19830 three cell type dataset.

The three cell type dataset contains measurements of 31099 genes on 42 samples. The cells correspond to a heterogeneous mixture of liver, brain and lung cells. The true proportions are given in Table 2.

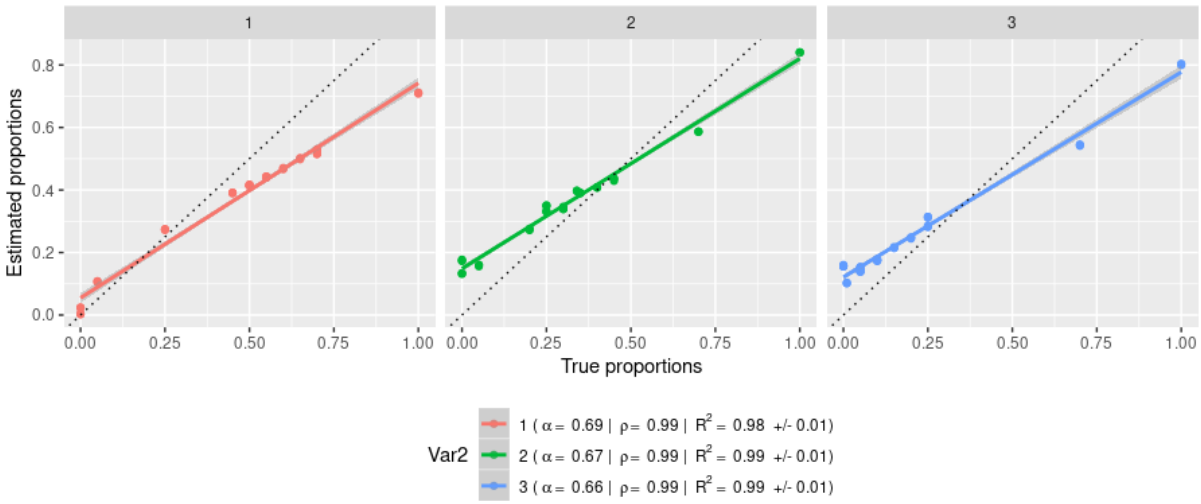

Figure 3: Plot of estimated versus true mixture proportions for the three cell types dataset using the Deconf method. (Left) Liver cells. (Middle) Brain cells. (Right) Lung cells. The algorithm converged after 4 iterations with an elapsed time of 0.79 seconds, giving a squared correlation coefficient of 0.99.

5 Discussion

Through attempting to replicate the results of Ogundijo and Wang (2017) and comparing the proposed SMC algorithm to existing methods, we concluded that the SMC algorithm was not exactly suitable for gene expression deconvolution. In our experiences using the SMC sampler, we encountered numerous disadvantages and inconsistencies. When using microarray gene expression data on a log_2 scale, as it is often expressed, estimates obtained were nonsensical. The computation time, while it can be improved through parallelisation, was excessive, particularly when compared to methods such as NMF. We were also concerned by the choice of Gaussian priors for parameters which were constrained to be positive or to lie in the interval $[0, 1]$. On more than one occasion, we obtained negative estimates and estimates greater than 1 for the cell proportions. Some further limitations of the Ogundijo and Wang (2017) paper stem from the limited discussion of the data preprocessing steps, as well as assuming independence of the data points to simplify the likelihood. While this assumption simplifies the analytical expressions for the sequence of target distributions, it is rarely the case that gene expression data is independent and we can expect there to be subsets of genes whose expressions are correlated with one another.

Furthermore, while Ogundijo and Wang (2017) acknowledge that one of the main advantages of SMC is its ability to be parallelised, the authors fail to look into ways to parallelise the algorithm. In the future, it would be useful to implement this and examine how the results and computation time are affected. Another area for further research would be to explore why data transformations significantly affected the performance of the algorithm. By understanding why this occurs, we may be able to identify which transformations lead to optimal algorithm performance. It would also be interesting to study the effect of varying the priors and imposing constraints on the sampled parameters. While the existing priors are conjugate with respect to the assumed form of the likelihood, we do feel that more appropriate prior distributions could lead to reduced computation time and improved parameter estimates. Finally, to further reduce computation time, we could examine gene selection techniques for identifying more informative subsets of genes.

References

Affymetrix (2009). https://www.thermofisher.com/uk/en/home/life-science/microarray-analysis/microarray-data-analysis/microarray-analysis-sample-data/gene-st-array-data-set.html. Accessed: 21-02-2019.

Doucet, A., De Freitas, N. and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pp. 3–14. Springer.

Erkkilä, T., Lähdesmäki, H., Shmulevich, I., Ruusuvuori, P., Lehmusvaara, S. and Visakorpi, T. (2010). Probabilistic analysis of gene expression measurements from heterogeneous tissues. *Bioinformatics*, **26**(20), 2571–2577.

Gaujoux, R. and Seoighe, C. (2010). A flexible R package for nonnegative matrix factorization. *BMC Bioinformatics*, **11**(1), 367. doi:10.1186/1471-2105-11-367. URL http://www.biomedcentral.com/1471-2105/11/367.

Gaujoux, R. and Seoighe, C. (2013a). CellMix: A Comprehensive Toolbox for Gene Expression Deconvolution. *Bioinformatics (Oxford, England)*, **29**. doi:10.1093/bioinformatics/btt351.

Gaujoux, R. and Seoighe, C. (2013b). Fast Combinatorial Non-Negative Least-Square. https://r-forge.r-project.org/scm/viewvc.php/*checkout*/www/CellMix/gedAlgorithm.deconf.html?revision=2&root=cellmix. Accessed: 18-02-2019.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U. and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**(2), 249–264.

Kim, H. and Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, **23**(12), 1495–1502.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**(6755), 788.

Nguyen, T. L. T., Septier, F., Peters, G. W. and Delignon, Y. (2016). Efficient sequential Monte-Carlo samplers for Bayesian inference. *IEEE Transactions on Signal Processing*, **64**(5), 1305–1319.

Ogundijo, O. E. and Wang, X. (2017). A sequential Monte Carlo approach to gene expression deconvolution. *PloS one*, **12**(10), e0186167.

Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, **5**(2), 111–126.

Picchini, U. (2016). Sequential Monte Carlo and the bootstrap filter. https://umbertopicchini.wordpress.com/2016/10/19/sequential-monte-carlo-bootstrap-filter/. Accessed: 18-02-2019.

Renaud Gaujoux, C. S. (2018). Complete Gene Expression Deconvolution: Method deconf. https://r-forge.r-project.org/scm/viewvc.php/*checkout*/www/fcnnls.html?revision=85&root=nmf. Accessed: 18-02-2019.

Repsilber, D., Kern, S., Telaar, A., Walzl, G., Black, G. F., Selbig, J., Parida, S. K., Kaufmann, S. H. and Jacobsen, M. (2010). Biomarker discovery in heterogeneous tissue samples-taking the in-silico deconfounding approach. *BMC bioinformatics*, **11**(1), 27.

Van Benthem, M. H. and Keenan, M. R. (2004). Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *Journal of Chemometrics: A Journal of the Chemometrics Society*, **18**(10), 441–450.

A Derivation of Target Densities

Here we present the derivation of the target densities for the cell-type specific expressions.

$$
\begin{aligned}
\pi_t(x_{ik}|\cdot) &\propto p(x_{ik}|\mu_{ik},\nu_{ik}) \left[\prod_{j=1}^{J} p(y_{ij}|\sum_{k'=1}^{K} x_{ik'}m_{k'j},\lambda)\right]^{\epsilon_t} \\
&\propto \exp\left[-\frac{\nu_{ik}}{2}(x_{ik}-\mu_{ik})^2 - \epsilon_t\lambda\sum_{j=1}^{J}\left(y_{ij}-\sum_{k'=1}^{K}x_{ik'}m_{k'j}\right)^2\right] \\
&\propto \exp\left\{-\frac{1}{2}\left[x_{ik}^2\left(\nu_{ik}+\epsilon_t\lambda\sum_{j=1}^{J}m_{kj}^2\right) - 2x_{ik}\left(\mu_{ik}\nu_{ik}+\epsilon_t\lambda\sum_{j=1}^{J}y_{ij}m_{kj} - \epsilon_t\lambda\sum_{j=1}^{J}y_{ijk}m_{kj}\right)\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\left(x_{ik}^2A_{ik}^t - 2x_{ik}B_{ik}^t\right)\right\} \\
&\propto \exp\left\{-\frac{A_{ik}^t}{2}\left[x_{ik}^2 - 2x_{ik}\frac{B_{ik}^t}{A_{ik}^t} + \left(\frac{B_{ik}^t}{A_{ik}^t}\right)^2\right]\right\} \\
&= \exp\left\{-\frac{A_{ik}^t}{2}\left[x_{ik}-\frac{B_{ik}^t}{A_{ik}^t}\right]^2\right\}.
\end{aligned}
$$

Thus, $\pi_t(x_{ik}|\cdot) \sim \mathcal{N}\Big(\frac{B_{ik}^t}{A_{ik}^t}, \frac{1}{A_{ik}^t}\Big)$, where $\mathcal{Y}_{ijk} = \sum_{k'\neq k} x_{ik'}m_{k'j}$, $A_{ik}^t = \nu_{ik} + \epsilon_t\lambda\sum_{j=1}^{J} m_{kj}^2$ and $B_{ik}^t = \mu_{ik}\nu_{ik} + \epsilon_t\lambda\Big(\sum_{j=1}^{J} y_{ij}m_{kj} - \sum_{j=1}^{J}\mathcal{Y}_{ijk}m_{kj}\Big)$.