A Sequential Monte Carlo Approach to Gene Expression Deconvolution

Emmanuelle Dankwa, Natalia Garcia Martin, William Thomas, Yuxi Jiang

OxWaSP

October 18, 2019

Sequential Importance Sampling

State-space Model

- Hidden states $\{x_t : t \in \mathbb{N}\}$
  - Markov process
  - Initial distribution $p(x_0)$
  - Transition probability $p(x_t|x_{t-1})$
- Observations $\{y_t : t \in \mathbb{N}\}$
  - Conditionally independent given $\{x_t : t \in \mathbb{N}\}$
  - Marginal distribution $p(y_t|x_t)$
- Posterior distribution $p(x_{0:t}|y_{1:t})$

Importance Sampling Algorithm

- Importance weights

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

- Not suitable for recursive estimation

Sequential Importance Sampling

- Proposal distribution

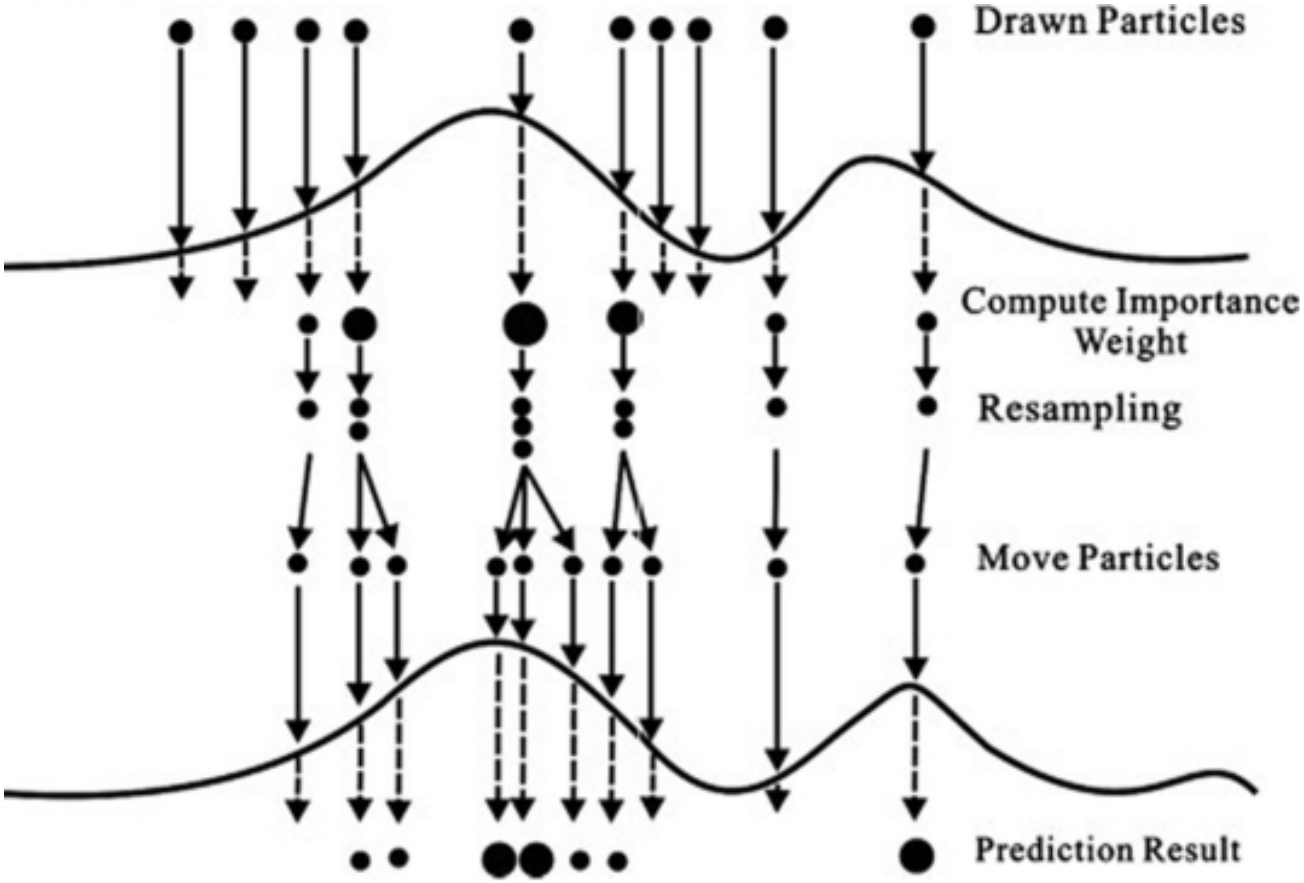
$$\begin{aligned}\pi(x_{0:t}|y_{1:t}) &= \pi(x_t|x_{0:t-1}, y_{1:t})\pi(x_{0:t-1}|y_{1:t-1}) \\ &= \pi(x_0)\prod_{k=1}^{t}\pi(x_t|x_{0:k-1}, y_{1:k}).\end{aligned}$$

- Importance weights

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

- Suffers from particle degeneracy

Resampling



[1](Picchini, 2016)

Bootstrap Filter

1. At time $t = 1$:
   - Draw samples $x_0^1, \ldots, x_0^N$ with $x_0^n \sim p(x_0)$
   - Assign to each sample the weights $\tilde{w}_0^n = 1/N, \;\; n = 1, \ldots, N$
2. At times $2 \leq t \leq T$
   1. Importance sampling step
      - Draw samples $\tilde{x}_t^1, \ldots, \tilde{x}_t^N$ with $\tilde{x}_t^n \sim p(x_t | x_{t-1}^n)$ and set $\tilde{x}_{0:t}^n = (\tilde{x}_{0:t-1}^n, \tilde{x}_t^n)$
      - Compute the unnormalised importance weights: $\tilde{w}_t^n = p(y_t | \tilde{x}_t^n)$.
      - Normalise the weights by $w_t^n = \frac{\tilde{w}_t^n}{\sum_{l=1}^N \tilde{w}_t^l}, \;\; n = 1, \ldots, N.$
   2. 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.

The Model I (Ogundijo and Wang, 2017)

- 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.
- The expression level of gene i in sample j is given by the sum of its expression across the K cell types.
- Assuming a linear relationship between the expression value of pure and mixed samples, we have:

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

where

x_{ik} : the expression of gene i in cell type k,

m_{kj}: the proportion of cell type k in sample j,

e_{ij}: additive Gaussian noise with zero mean and variance, λ^{-1}.

The Model II (Ogundijo and Wang, 2017)

- In matrix form : $\underbrace{\boldsymbol{Y}}_{I\times J} = \underbrace{\boldsymbol{X}}_{I\times K}\underbrace{\boldsymbol{M}}_{K\times J} + \underbrace{\boldsymbol{E}}_{I\times J}$.
- Goal: To infer $\boldsymbol{X}$, $\boldsymbol{M}$ and λ given $\boldsymbol{Y}$.
- Normality assumption:

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

- Assuming i.i.d measurements, the joint likelihood is given by:

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

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.

Parameter Estimation

Idea:

- Difficult to sample directly from $p(\boldsymbol{\theta}|\boldsymbol{Y})$.
- Introduce sequence of intermediate target distributions, $\{\pi_t\}_{t=1}^{T}$ such that $p(\boldsymbol{\theta}) = \pi_1$ and $p(\boldsymbol{\theta}|\boldsymbol{Y}) = \pi_T$.
- $\{\pi_t\}_{t=1}^{T}$ satisfies the expression

$$\pi_t(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta})p(\boldsymbol{Y}|\boldsymbol{\theta})^{\epsilon_t},$$

where $\{\epsilon_t\}_{t=1}^{T}$ is defined as a non-decreasing temperature schedule with $\epsilon_1 = 0$ and $\epsilon_T = 1$ and serves to gradually introduce the effect of the likelihood.

Densities of Model Parameters I

1. Cell-type specific expression
   - **Conjugate Prior**: $x_{ik} \sim \mathcal{N}(\mu_{ik}, \nu_{ik}^{-1})$, where both μ_{ik} and ν_{ik}^{-1} are known.
   - **Target density**: $\pi_t(x_{ik}|\cdot) \sim \mathcal{N}\left(\frac{B_{ik}^t}{A_{ik}^t}, \frac{1}{A_{ik}^t}\right)$, where

$$A_{ik}^t = \nu_{ik} + \epsilon_t \lambda \sum_{j=1}^{J} m_{kj}^2 \quad \text{and} \quad B_{ik}^t = \mu_{ik}\nu_{ik} + \epsilon_t \lambda \sum_{j=1}^{J} (y_{ij} m_{kj} - \mathcal{Y}_{ijk} m_{kj}).$$

Densities of Model Parameters I

1. Cell-type specific expression
   - **Conjugate Prior**: $x_{ik} \sim \mathcal{N}(\mu_{ik}, \nu_{ik}^{-1})$, where both μ_{ik} and ν_{ik}^{-1} are known.
   - **Target density**: $\pi_t(x_{ik}|\cdot) \sim \mathcal{N}\left(\frac{B_{ik}^t}{A_{ik}^t}, \frac{1}{A_{ik}^t}\right)$, where

$$A_{ik}^t = \nu_{ik} + \epsilon_t \lambda \sum_{j=1}^{J} m_{kj}^2 \quad \text{and} \quad B_{ik}^t = \mu_{ik}\nu_{ik} + \epsilon_t \lambda \sum_{j=1}^{J} (y_{ij} m_{kj} - \mathcal{Y}_{ijk} m_{kj}).$$

2. Cell type proportions
   - **Conjugate Prior**: $m_{kj} \sim \mathcal{N}(\mu_{kj}, \nu_{kj}^{-1})$
   - **Target density**: $\pi_t(m_{kj}|\cdot) \sim \mathcal{N}\left(\frac{V_{kj}^t}{U_{kj}^t}, \frac{1}{U_{kj}^t}\right)$, where

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

Densities of Model Parameters II

3. Precision
   - **Conjugate Prior**: $\lambda \sim \text{Gamma}(\alpha, \beta)$
   - **Target density**: $\pi_t(\lambda|\cdot) \sim \text{Gamma}(\hat{\alpha}, \hat{\beta})$, where

$$\hat{\alpha} = \alpha + \frac{\epsilon_t IJ}{2} \quad \text{and} \quad \hat{\beta} = \beta + \frac{\epsilon_t}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} \left(y_{ij} - \sum_{k=1}^{K} x_{ik} m_{kj} \right)^2 .$$

SMC Algorithm for Gene Deconvolution I

1. Input the heterogeneous gene expression matrix $\boldsymbol{Y}$, the prior parameters and the temperature schedule $\{\epsilon_t\}_{t=1}^{T}$.
2. Set $t = 1$.

for $n = 1$ to N **do**
  draw a sample from $\texttt{Gamma}\ (\alpha,\ \beta)$
  for $k = 1$ to K; $j = 1$ to J **do**
    draw a sample from $\mathcal{N}(\mu_{kj},\ \nu_{kj}^{-1})$
  end for
  for $i = 1$ to I; $k = 1$ to K **do**
    draw a sample from $\mathcal{N}(\mu_{ik},\ \nu_{ik}^{-1})$
  end for
end for

Set $w_1^n = 1/N,\ n = 1, \ldots, N$.

SMC Algorithm for Gene Deconvolution II

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

end for

SMC Algorithm for Gene Deconvolution III

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 $\hat{\boldsymbol{\theta}}$.

Two cell type example using SMC

- Gene 1.0 ST Array Data Set from Affymetrix (2009).
- 33 samples of human heart and brain tissue.
- 6 pure samples, 27 heterogeneous samples.
- Expression values for 33,297 genes.

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

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

Two cell type example using SMC

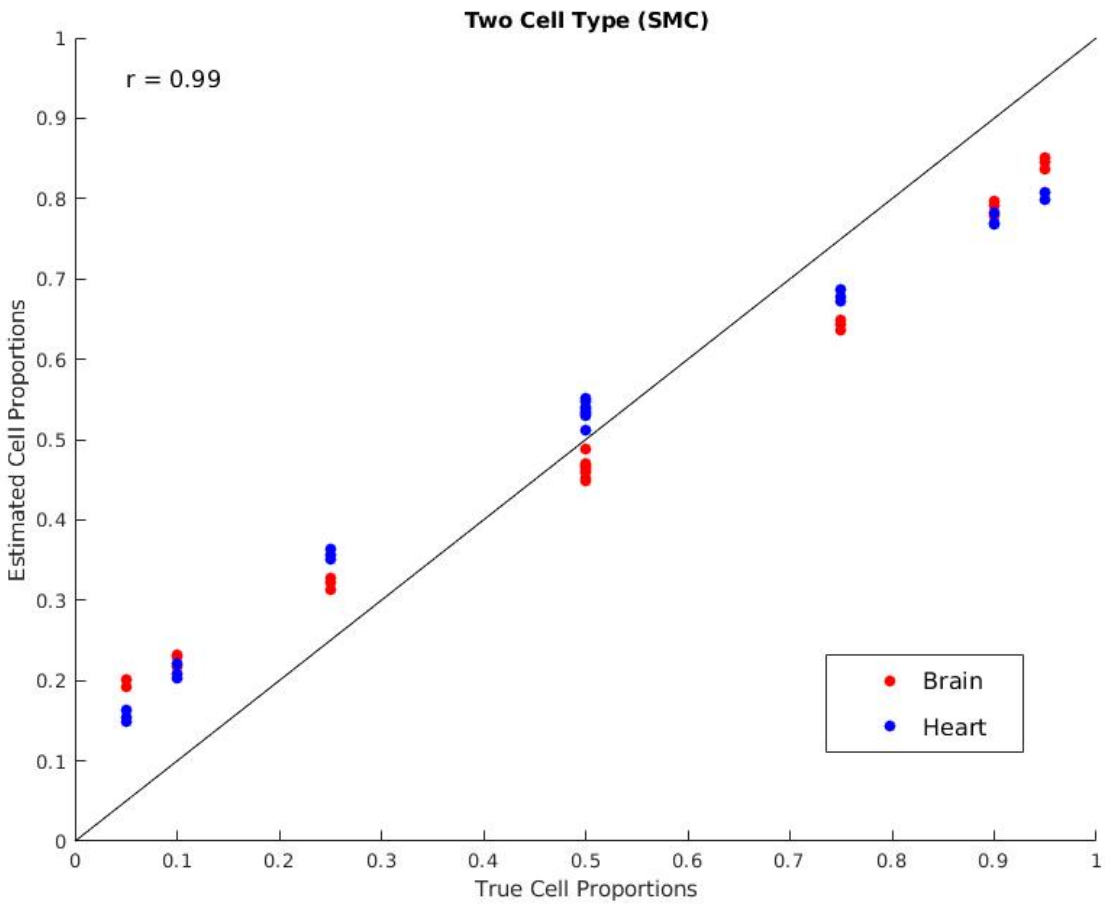


Figure: Plot of estimated versus true mixture proportions for the two cell types dataset using the proposed SMC method. 2000 randomly selected genes were used, with T = 5000 and N = 40. Run time was 3.36 hours.

Two cell type example using NMF

- CellMix
- DSection - MCMC
- Deconf - NMF

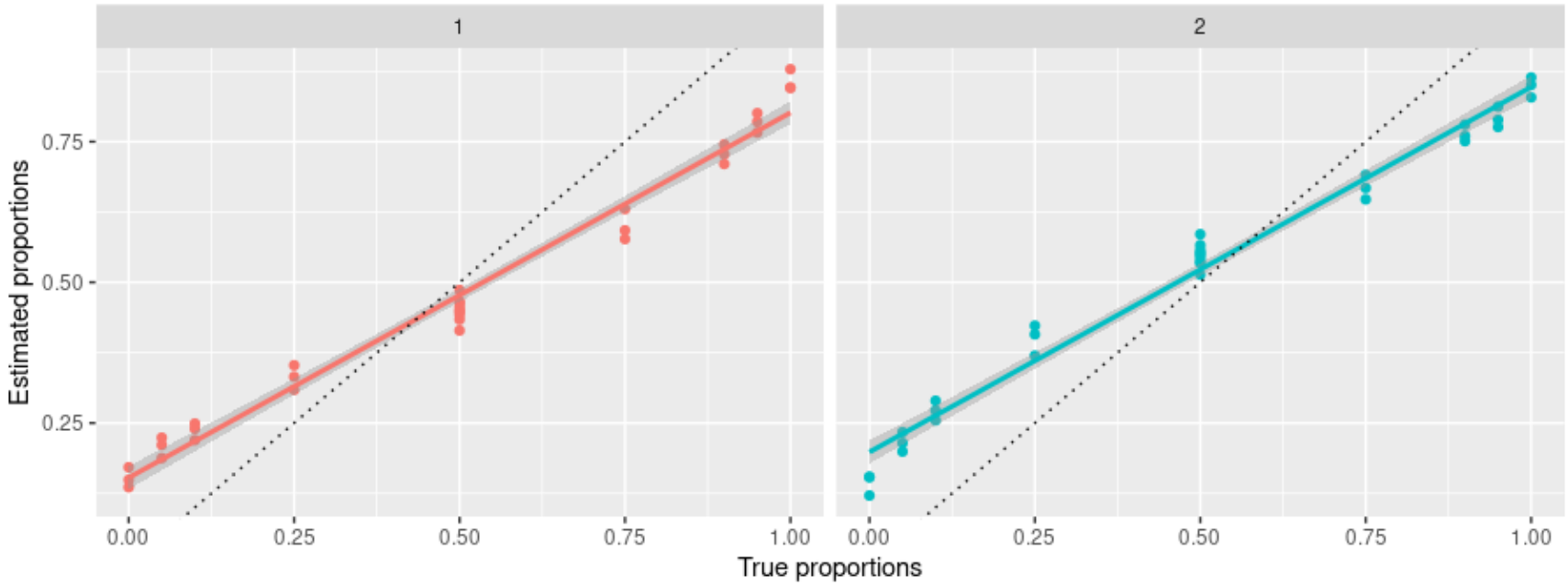


Figure: Plot of estimated versus true mixture proportions for the 2 cell types dataset using the Deconf method. (Left) Brain cells. (Right) Heart cells. The algorithm converged after 8 iterations with an elapsed time of 0.998 seconds, giving a squared Pearson correlation coefficient of 0.98.

Deconf method (Repsilber et al., 2010)

Recall $\boldsymbol{Y} = \boldsymbol{XM} \quad (+\boldsymbol{E})$

- $\boldsymbol{Y}$: gene expression across samples, I x J
- $\boldsymbol{X}$: gene expression across cell types, I x K
- $\boldsymbol{M}$: cell type proportions across samples, K x J

  ($\boldsymbol{E}$: Gaussian noise with mean zero and precision λ)

Constraints

- $\boldsymbol{X}$ is non-negative and has been normalised (certered or quantile normalisation)
- Entries of $\boldsymbol{M}$ between 0 and 1
- Columns of $\boldsymbol{M}$ sum to one

Non-negative matrix factorisation (NMF)

- NMF idea: factorise a matrix into two other matrices $Y \approx XM$ with the property that all three matrices have no negative elements

$$\min_{X,M} ||Y - XM||_F \text{ s.t. } X, M \geq 0$$

- Applications: Latent features, high-dimensional data, clustering
- New implementation using *fcnnls*: Fast Combinatorial Nonnegative Least-Squares algorithm (van Benthem and Keenan, 2004)

Deconf method (Repsilber et al., 2010)

Performs NMF using alternating nonnegative least squares

1. Normalise columns of $\boldsymbol{Y}$
2. Generate starting values for $\boldsymbol{X}$ and $\boldsymbol{M}$
3. Apply constraints to $\boldsymbol{X}$ and $\boldsymbol{M}$
4. Fixing $\boldsymbol{X}$, calculate $\boldsymbol{M}$ using *lsqnoneg* (least squares non-negative matrix factorization)
5. Apply constrains to $\boldsymbol{X}$
6. Fixing $\boldsymbol{M}$, calculate $\boldsymbol{X}$ using *lsqnoneg*
7. Apply constrains to $\boldsymbol{M}$
8. Repeat from (4). Stop when $|\boldsymbol{Y} - \boldsymbol{XM}| < a$ or number of iterations $> b$ (eg. $a = 0.1$, $b = 100$)

Three cell type example using NMF

- Liver, brain and lung
- 31099 genes, 42 samples

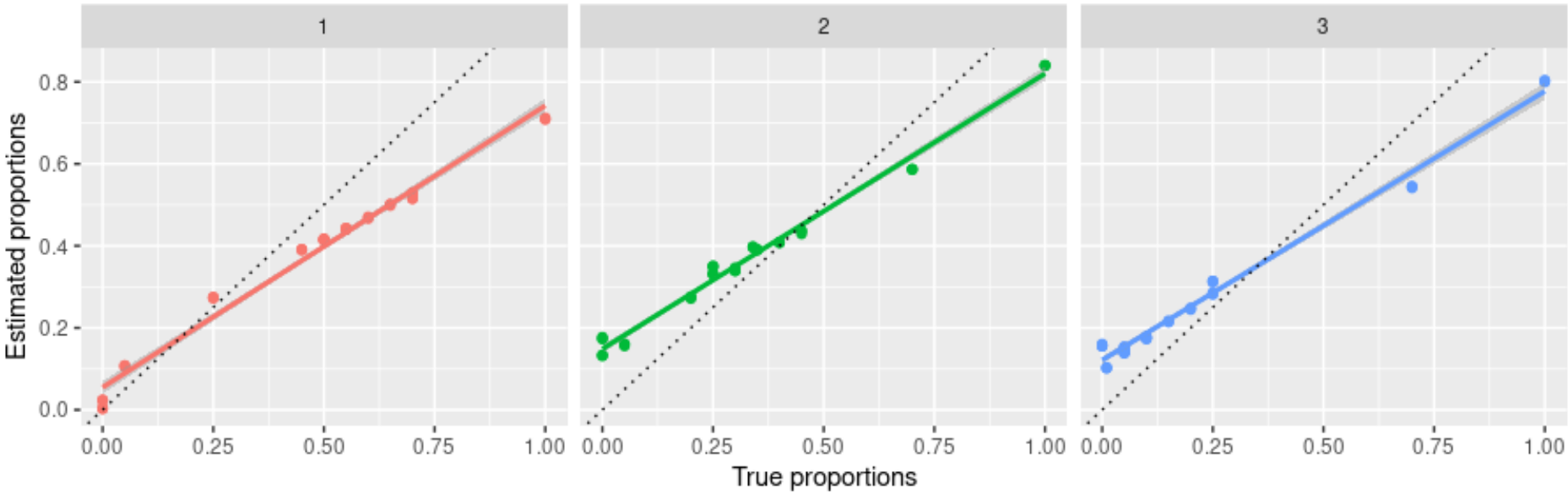


Figure: Plot of estimated versus true mixture proportions for the 3 cell types dataset using the Deconf method. (Left) Liver cells. (Middle) Brain cells. (Right) Lung cells. The algorithm converged after 4 iterations with an elapsed time of 0.79 seconds, giving a squared correlation coefficient of 0.99.

Discussion

Limitations

- Really poor estimates when using the $\log_2$ scale
- Excessive computation time
- Choice of Gaussian priors
- Negative proportion estimates
- Preprocessing steps not discussed clearly
- Independence of the data points to simplify the likelihood

Extensions

- Parallelisation
- Other prior choices
- Identification of more informative subsets of genes

References I

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

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

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

References II

Dirk Repsilber, Sabine Kern, Anna Telaar, Gerhard Walzl, Gillian F Black, Joachim Selbig, Shreemanta K Parida, Stefan HE Kaufmann, and Marc Jacobsen. Biomarker discovery in heterogeneous tissue samples-taking the in-silico deconfounding approach. *BMC bioinformatics*, 11(1):27, 2010.