THE UNIVERSITY OF BRITISH COLUMBIA

DEPARTMENT OF STATISTICS

TECHNICAL REPORT #236

On-line Changepoint Detection and Parameter Estimation for Genome-wide Transcript Analysis

Francois Caron, Raphael Gottardo and Arnaud Doucet

November 2007

On-line Changepoint Detection and Parameter Estimation for Genome-wide Transcript Analysis

François Caron†‡, Raphael Gottardo† and Arnaud Doucet†‡
†Department of Statistics,
University of British Columbia,
333-6356 Agricultural Road,
Vancouver, BC, V6T 1Z2, Canada.
‡Department of Computer Science,
University of British Columbia,
2366 Main Mall,
Vancouver, BC, V6T 1Z4, Canada.

November 27, 2007

Abstract

We consider the problem of identifying novel RNA transcripts using tiling arrays. Standard approaches to this problem rely on the calculation of a sliding window statistic or on simple changepoint models. These methods suffer from several drawbacks including the need to determine a threshold to label transcript regions and/or specify the number of transcripts. In this paper, we propose a Bayesian multiple changepoint model to simultaneously identify the number of transcripts, the transcript boundaries and their associated levels. We also present a computationally efficient on-line algorithm which allows us to jointly estimate both the changepoint locations and the model parameters. Using two publicly available transcription data sets, we compare our method to a common sliding window approach and a simple changepoint model. In addition, we also provide the results of a simulation study which shows that our on-line estimation procedure provides good estimates of transcript boundaries and model parameters.

KEY WORDS: Tiling arrays; Sequential Monte Carlo; Particle filtering; Recursive parameter estimation.

1 Introduction

The advent of microarray technology (Lockhart et al. 1996) has enabled biomedical researchers to monitor changes in the expression levels of thousands of genes. However, investigations into the mechanisms driving these changes have only recently attained success with this high-throughput data. Affymetrix (Santa Clara, CA), NimbleGen Systems (Madison, WI), and Agilent Technologies (Palo Alto, CA) have recently developed oligonucleotide arrays that tile all of the non-repetitive genomic sequences of humans and other eukaryotes. Currently, tiling arrays can contain up to 7 million probes of length 25-50 base pairs (bps), spanning the non-repetitive regions of a given genome at high resolution. For example, one Affymetrix tiling array is enough to cover the whole yeast genome at an average resolution of 8bps (David et al. 2006). These tiling arrays have great potential and can be used for numerous genome-wide studies including transcriptome analysis (Bertone et al. 2004; Cheng et al. 2005; David et al. 2006), DNA protein and chromatin modification assays (ChIP-chip) (Kapranov et al. 2002; Carroll et al. 2005) and DNA variation detection (Mockler et al. 2005). Given their high density (several million genomic sequences for small eukaryote genomes) and high noise-to-signal ratio, the power of tiling arrays continues to be limited by a lack of appropriate statistical tools.

In this paper, we focus on the detection of novel RNA transcripts from tiling arrays- a problem that has received little attention from the statistics community. A common approach is to combine probe measurements via a sliding window (SW) statistic computed over neighboring probes, and then to apply a thresholding on the resulting statistics to call transcript regions (Kapranov et al. 2002; Bertone et al. 2004; Cheng et al. 2005). The threshold itself can be derived using con-

trol regions or by making various distributional assumptions for the statistics. A difficulty with SW approaches is that the resulting statistics are not independent due to the fact that each statistic uses information from neighboring probes It is also challenging to derive a meaningful threshold without control regions. Another problem is that the window size, which is fixed and has to be determined in advance, can have a big impact on the final results. Too small of a window can lead to many false positives whereas too large of a window can lead to over smoothing and poor detections of the region's endpoints (Huber et al. 2006). As an alternative to SW approaches, Huber et al. (2006) described a segmentation algorithm to find a globally optimal fit of a piecewise constant expression profile along genomic coordinates. This algorithm is based on a simple changepoint model, which models the changes in transcription along genomic coordinates as piecewise constant. Even though Huber et al. (2006) showed that their changepoint model can provide more accurate segmentation than SW approaches, it still suffers from major pitfalls. The number of segments has to be fixed in advance, but this number is usually unknown. In addition, it is difficult to selectively threshold the segment transcription levels in order to call transcript regions.

In this paper, we introduce a Bayesian changepoint model to simultaneously identify transcript boundaries and their associated levels. Our model also builds on previous approaches used in gene expression analysis (Newton et al. 2001; Gottardo et al. 2003; Gottardo et al. 2006) and uses a mixture distribution to classify segments as transcript or non-transcript, hence no thresholding is necessary. Given the data, we are interested in inferring the changepoint locations, the number of changepoints, and the model parameters; these parameters have a large impact on the resulting changepoint segmentation and are typically unknown. In our context, due to the large number of features, it would be extremely expen-

sive to use Markov chain Monte Carlo (MCMC) methods. Moreover the strong correlation between the changepoint locations and the parameters would induce poor mixing, and thus slow convergence, for any Gibbs sampling type strategy. Even in the simplest scenario where all the parameters are known, performing exact Bayesian inference for the changepoints has a computational complexity quadratic in the number of features, which is too expensive for our target application. An alternative to MCMC are Sequential Monte Carlo methods, also known as particle filtering methods. Recently, a variant of the particle filtering approximation technique with computational complexity linear in the number of features was proposed in Fearnhead and Liu (2007). We propose here a new recursive maximum likelihood estimation procedure which allows us to jointly estimate both the changepoint locations and the model parameters. The exact version of this algorithm also has a computational complexity quadratic in the number of features but, coupled to a particle approximation in the spirit of Fearnhead and Liu (2007), we obtain an approximation whose computational cost is linear in the number of features. This scales well with advances in feature density and tiling array size.

Our paper is organized as follows, Section 2 introduces the data structure and the notation. In Section 3 and 4, we present our changepoint model and parameter estimation procedure, respectively. In Section 5, we apply our method to experimental data and compare it to the changepoint model of Huber et al. (2006) and the SW approach used by Cheng et al. (2005). Section 6 presents the results of a simulation study to demonstrate the performance of our on-line estimation method. Finally, in Section 7 we discuss our results and possible extensions. All the calculations are detailed in Appendices.

2 Data

We use two publicly available datasets to demonstrate our methodology. In the first, David et al. (2006) use high density Affymetrix tiling arrays with 25-mer oligonucleotides spaced every 4bps on average to interrogate both strands of the full *Saccharomyces cerevisiae* genome. We will refer to these data as the *yeast* data. In the second, Cheng et al. (2005) use tiling arrays to map the sites of transcription for approximately 30% of the human genome encoded in 10 human chromosomes (6, 7, 13, 14, 19, 20, 21, 21, X, and Y). Similar to David et al. (2006), Cheng et al. (2005) use Affymetrix high density tiling arrays with 25-mer oligonucleotides spaced every 5 bps on average. These data, which we will refer to as the *human* data, also contain experimentally verified transcripts which will allow us to validate our methodology.

Similar to oligonucleotide gene expression arrays (Lockhart et al. 1996), Affymetrix tiling arrays query each sequence of interest with a perfect match (PM) and a mismatch (MM) probe, where the MM probe is complementary to the sequence of interest except at the central base, which is replaced with its complementary base. The difference is that the probes used on tiling arrays do not necessarily belong to genes, which allows for an unbiased mapping of RNA transcripts. Following the idea that MM intensities are poor measures of non-specific hybridization (Irizarry et al. 2003), we only used the PM intensities. In the case of the yeast data, the data were normalized using the procedure of David et al. (2006) and described in Huber et al. (2006), which is part of the `tilingArray` package available from Bioconductor (Gentleman et al. 2004). In the case of the human data, the data were normalized by quantile normalization (Bolstad et al. 2003), as in Cheng et al. (2005). After normalization, the data take the form $\{y_{tr} : t = 1, \ldots, T; r = 1 \ldots R\}$, where

y_{tr} is the normalized intensity of probe (also called time in the following) t from replicate r. Here, we assume that the probes are ordered by genomic positions, where we denote by $\{x_t : t = 1, \ldots, T\}$ the corresponding positions arranged in increasing order, that is $x_t < x_{t'}$ for $t < t'$. Finally, we will summarize each probe measurement by the mean of its normalized intensities across replicates, and we will denote the resulting summaries by $\{z_t : t = 1, \ldots, T\}$. Such summaries are often used in microarray studies to facilitate modeling, reduce the computational burden, and avoid across-array normalization issues; see for example Efron (2004) and Do et al. (2005).

3 Statistical Model

Using the notation introduced in Section 2, we will denote by $z_{t_1:t_2} = \{z_{t_1}, \ldots, z_{t_2}\}$ the vector of intensities from t_1 to t_2. As in Huber et al. (2006), we consider a changepoint model in which the changes in transcription along genomic coordinates can be modeled as piecewise constant. It follows that the sequence of observations $z_1, \ldots, z_T$ can be partitioned into $m + 1$ contiguous segments $z_{1:\tau_1}, z_{\tau_1+1:\tau_2}, \ldots, z_{\tau_m:T}$ where the index $\tau_1, \ldots, \tau_m$ are called the changepoints, m being unknown. A Bayesian changepoint model is defined by a joint distribution over the changepoints and the data. We consider changepoint models for the data with the following conditional independence assumption (Barry and Hartigan 1992; Fearnhead and Liu 2007): "*given the position of a changepoint, the data before that changepoint is independent of the data after the changepoint*". The number of changepoints is unknown as are their positions. The changepoint positions are modeled as a Markov process

$$\Pr(\text{“next changepoint at } t_2\text{”} \,|\, \text{“changepoint at } t_1\text{”}) = h(t_2 - t_1) \qquad (1)$$

i.e. the probability of a changepoint depends on the index distance to the previous one. This model is a special case of a product partition model for changepoints (Barry and Hartigan 1992; Fearnhead and Liu 2007). In our case, the function h is chosen to be a negative binomial distribution with parameters ρ and d, such that

$$\begin{aligned} h(x) &= \text{Negbin}(x-u;\rho,d) \\ &= \begin{cases} \frac{\Gamma(d+x-u)}{\Gamma(x-u+1)\Gamma(d)}\rho^d(1-\rho)^{x-u} & \text{if } x \geq u \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where u controls the smallest distance between two changepoints, d controls the shape of the distribution, which has a mode greater than u for $d > 1$, and ρ controls the average length of the segments. In the two examples explored in this paper we will fix u to 15, corresponding to a minimum segment size of approximately 100bps, and d to 2, allowing for a positive mode. We estimate ρ using the algorithm presented in Section 4. Finally, we denote by H, $H(l) = \sum_{i=1}^{l} h(i)$ the cumulative distribution associated with h, which will be used in Section 4 when we describe our estimation procedure.

Let $\tau_1 < \tau_2 < \ldots < \tau_m$ be the m successive unknown changepoints and set $\tau_0 = 0$ and $\tau_{m+1} = T$. The changepoints define $m+1$ segments, with segment i consisting of observations $z_{\tau_i+1:\tau_{i+1}}$, for $i = 0, \ldots, m$. We assume that each segment may be either *transcribed* or *non-transcribed*. Let us denote by λ the probability that a segment is transcribed and $r_i \in \{0, 1\}$ the associated latent variable indicating if segment i is transcribed ($r_i = 1$) or not ($r_i = 0$). It follows that $r_i \sim \text{Ber}(\lambda)$, that is Bernoulli with parameter λ. If $r_i = 1$ (transcript), then the data $z_{\tau_i+1:\tau_{i+1}}$, are assumed to be distributed from a normal/normal-inverse

gamma compound distribution, as follows,

$$(z_t|\mu_i, \sigma_i^2, r_i = 1) \sim \mathcal{N}(\mu_i, \sigma_i^2) \text{ for } t = \tau_i + 1, \ldots \tau_{i+1} \tag{2}$$
$$(\mu_i, \sigma_i^2|r_i = 1) \sim \mathcal{N}i\mathcal{G}(m_1, s_1, \nu_1, \gamma_1)$$

where $\mathcal{N}i\mathcal{G}(m_1, s_1, \nu_1, \gamma_1)$ is the normal-inverse gamma distribution, defined in Appendix A, with parameters m_1, s_1, ν_1 and γ_1 and $\mathcal{N}(\mu_i, \sigma_i^2)$ is the normal distribution with mean μ_i and variance σ_i^2. If $r_i = 0$ (not a transcript), the data $z_{\tau_i+1:\tau_{i+1}}$, are assumed to arise from a mixture of a skew t-distribution and a normal-normal inverse gamma compound distribution. If we introduce another latent variable $q_i \sim$Ber(p_0), $p_0 \in [0, 1]$, we can write

$$(z_t|r_i = 0, q_i = 1) \sim \mathcal{N}(\mu_i, \sigma_i^2) \text{ for } t = \tau_i + 1, \ldots \tau_{i+1} \tag{3}$$
$$(\mu_i, \sigma_i^2|r_i = 0, q_i = 1) \sim \mathcal{N}i\mathcal{G}(m_0, s_0, \nu_0, \gamma_0)$$

and

$$(z_t|r_i = 0, q_i = 0) \sim st(\varphi_0, \psi_0, \zeta_0, \xi_0)$$

where m_0, s_0, ν_0, γ_0, φ_0 and ψ_0 are unknown parameters that will be estimated while ζ_0 and ξ_0 will be fixed in advanced. Note that in (3) resp. (2), the unknown parameters are shared across non-trancript segments, resp. transcripts, which allows us to borrow strength across segments when estimating segment boundaries. The skew t-distribution, $st(\varphi_0, \psi_0, \zeta_0, \xi_0)$, is as defined in Azzalini and Capitanio (2003) and whose density is given in Appendix A. The parameters φ_0, ψ_0, ζ_0 and ξ_0 represent the location, scale, degrees of freedom, and skewness parameters. In the example explored in this paper, we will use $\zeta_0 = 4$ for the degrees of freedom parameter to provide for robustness against outliers, and $\xi_0 = 10$ for the skewness parameter, which seems to be enough to deal with the skewness observed for non-transcript segments. Even though these parameters could be

estimated, we have chosen to fix them for simplicity. However, the exact value of these parameters is not crucial; experimentation showed that different values give similar results. For the non-transcribed segments, we have found it necessary to introduce a skew t-distribution to deal with frequent outliers and the skewed nature of low-intensity observations. We have experimented with a single normal/normal-inverse gamma compound distribution for the baseline and the results were not as good in terms of goodness of fit and segmentation results (data not shown). In order to have the same mean value for the non-transcribed segments, we assume that the mean of the skew t-distribution is equal to m_0, that is we set $m_0 = \varphi_0 + \psi_0\xi_0/\sqrt{1+\xi_0^2}\sqrt{\zeta_0/\pi}\Gamma((\zeta_0-1)/2)/\Gamma(\zeta_0/2)$. Note that we use the same generic variables μ_i and σ_i^2 in (2) and (3) for ease of notation even though these are different parameters. In any case, these variables are nuisance parameters which will be integrated out later on.

Conditioning on two consecutive changepoints τ_i and τ_{i+1} and the unknown parameters, which are omitted below for ease of notation, the marginal likelihood is given by

$$
\begin{aligned}
P(\tau_i, \tau_{i+1}) &= p(z_{\tau_i+1:\tau_{i+1}}) \qquad (4)\\
&= (1-\lambda)p(z_{\tau_i+1:\tau_{i+1}}|r_i=0) + \lambda p(z_{\tau_i+1:\tau_{i+1}}|r_i=1)\\
&= (1-\lambda)\left[\begin{array}{l}(1-p_0)\prod_{t=\tau_i+1}^{\tau_{i+1}} st(z_t|\varphi_0,\psi_0,\zeta_0,\xi_0)+\\ p_0 \iint \prod_{t=\tau_i+1}^{\tau_{i+1}} \mathcal{N}(z_t|\mu_i,\sigma_i^2)\mathcal{N}i\mathcal{G}(\mu_i,\sigma_i^2|m_0,s_0,\nu_0,\gamma_0)d\mu_i d\sigma_i^2\end{array}\right]\\
&+ \lambda \iint \prod_{t=\tau_i+1}^{\tau_{i+1}} \mathcal{N}(z_t|\mu_i,\sigma_i^2)\mathcal{N}i\mathcal{G}(\mu_i,\sigma_i^2|m_1,s_1,\nu_1,\gamma_1)d\mu_i d\sigma_i^2
\end{aligned}
$$

where the integrals can be computed analytically (see Appendix B).

4 Changepoint Detection and Parameter Estimation

4.1 Exact inference

4.1.1 Filtering recursions

Let C_t denote the time of the most recent changepoint prior to t (with $C_t = 0$ if there has been no changepoint before time t). Conditional on $C_{t-1} = j$, either $C_t = j$, i.e. there is no changepoint at time t, or $C_t = t-1$ if there is a changepoint. The transition probabilities are

$$f(C_t = j | C_{t-1} = i) = \begin{cases} \frac{1-H(t-i-1)}{1-H(t-i-2)} & \text{if } j = i \\ \frac{H(t-i-1)-H(t-i-2)}{1-H(t-i-2)} & \text{if } j = t-1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and

$$g(z_t | C_t = j, z_{1:t-1}) = \begin{cases} \frac{P(j,t)}{P(j,t-1)} & \text{if } j < t-1 \\ P(t-1,t) & \text{if } j = t-1 \end{cases}$$

where $P(\cdot,\cdot)$ is given by (4). The so-called filtering distributions $p(C_t|z_{1:t})$ can be computed recursively in time using Fearnhead and Liu (2007)

$$p(C_t | z_{1:t}) = \frac{\xi(C_t, z_{1:t})}{\sum_{i=0}^{t-1} \xi(C_t = i, z_{1:t})}$$

where we denote

$$\xi(C_t, z_{1:t}) = g(z_t | C_t, z_{1:t-1}) p(C_t | z_{1:t-1})$$

which satisfies the following recursion

$$\xi(C_t, z_{1:t}) = \begin{cases} g(z_t | C_t = j, z_{1:t-1}) f(C_t = j | C_{t-1} = j) p(C_{t-1} = j | z_{1:t-1}) & \text{if } j < t-1 \\ g(z_t | C_t = j, z_{1:t-1}) \sum_{i=0}^{t-2} f(C_t = j | C_{t-1} = i) p(C_{t-1} = i | z_{1:t-1}) & \text{if } j = t-1, \end{cases} \tag{6}$$

Once the filtering distributions $p(C_t|z_{1:t})$ are stored for all $t = 1, \ldots, n$, we can simulate from the joint posterior distribution of the changepoints before time n (Chopin 2007; Fearnhead and Liu 2007), as follows.

Simulation of changepoints from the joint posterior distribution

- Simulate τ_1 from $p(C_n|z_{1:n})$. Set $k = 1$.
- While $\tau_k > 0$
  - Sample τ_{k+1} proportionally to $f(C_{\tau_k+1} = \tau_k|C_{\tau_k})p(C_{\tau_k}|z_{1:\tau_k})$ and set $k = k+1$.

4.1.2 MAP recursions

An on-line Viterbi algorithm can be designed for calculating the maximum a posteriori (MAP) estimate of the changepoints and model orders (Fearnhead and Liu 2007). Let $\mathcal{M}_j$ be the event that given a changepoint at time j, the MAP estimate of changepoints and model has occurred prior to time j. Then for $t = 1, \ldots, n$, $j = 0, \ldots, t-1$ and $r = 0, 1$ (non transcribed or transcribed segment), we define

$$P_t(j, r) = \Pr(C_t = j, \text{model } r, \mathcal{M}_j, z_{1:t}),$$
$$P_t^{MAP} = \Pr(\text{Changepoint at } t, \mathcal{M}_t, z_{1:t}).$$

At time t, the MAP estimate $\widehat{c}_t$ of C_t and the current model are given by the values of j and r which maximise $P_t(j, r)$. The following recursions can be established

$$\begin{aligned} P_t(j, r) &= (1 - H(t - j - 1))P(j, t|\, r) \Pr(r) P_j^{MAP}, \\ P_t^{MAP} &= \max_{j,r} \left(\frac{P_t(j, r)h(t - j)}{1 - H(t - j - 1)} \right) \end{aligned} \tag{7}$$

where $P(j, t|r)$ is the marginal distribution of the observations $z_{j+1:t}$ assumed to be in the same segment following the model r.

4.1.3 Recursive parameter estimation

The previous recursions assume the parameters $\boldsymbol{\theta} = \{p_0, \psi_0, m_0, s_0, \nu_0, \gamma_0, \lambda, m_1, s_1, \nu_1, \gamma_1, \rho\}$ are known. However, these parameters have a strong influence on the changepoint estimates. Here, we propose to estimate them using a recursive maximum likelihood approach. We introduce a subscript $\boldsymbol{\theta}$ to emphasize

the dependence on parameters $\boldsymbol{\theta}$ of the filtering density $p_{\boldsymbol{\theta}}(C_t|z_{1:t})$, the transition probability $f_{\boldsymbol{\theta}}(C_t|C_{t-1})$, the conditional predictive density $g_{\boldsymbol{\theta}}(z_t|C_t, z_{1:t-1})$ and $\xi_{\boldsymbol{\theta}}(C_t, z_{1:t}) = g_{\boldsymbol{\theta}}(z_t|C_t, z_{1:t-1})p_{\boldsymbol{\theta}}(C_t|z_{1:t-1})$.

The log-likelihood conditional on the data $z_{1:t}$ is given by

$$l_t(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(z_1) + \sum_{k=2}^{t} \log p_{\boldsymbol{\theta}}(z_k|z_{1:k-1}) \tag{8}$$

where

$$p_{\boldsymbol{\theta}}(z_t|z_{1:t-1}) = \sum_{j=0}^{t-1} \xi_{\boldsymbol{\theta}}(C_t = j, z_{1:t}). \tag{9}$$

As $t \to \infty$, we have

$$\lim_{t\to\infty} \frac{l_t(\boldsymbol{\theta})}{t} = l(\boldsymbol{\theta}).$$

This follows from the fact that (1)-(2)-(3) define an (asymptotically) stationary process with 'good' mixing properties. Moreover, $l(\boldsymbol{\theta})$ admits the true parameter $\boldsymbol{\theta}^*$ as a global maximum. To find a local maximum of $l(\boldsymbol{\theta})$, we use a stochastic approximation algorithm (Benveniste et al. 1990)

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \gamma_t \nabla \log p_{\boldsymbol{\theta}_{1:t-1}}(z_t|z_{1:t-1}) \tag{10}$$

where the stepsize sequence $\{\gamma_t\}$ is a positive non-increasing sequence such that $\sum \gamma_t = \infty$ and $\sum \gamma_t^2 < \infty$ whereas $\nabla \log p_{\boldsymbol{\theta}_{1:t-1}}(z_t|z_{1:t-1})$ is the gradient of the predictive log-likelihood. The subscript $\boldsymbol{\theta}_{1:t-1}$ indicates that this predictive likelihood is computed using $\boldsymbol{\theta} = \boldsymbol{\theta}_k$ at time $k+1$. Under regularity conditions (Benveniste et al. 1990), it can be shown that $\boldsymbol{\theta}_t$ will converge to a local maximum of $l(\boldsymbol{\theta})$. To improve the convergence rate of this algorithm, we can also use a Newton or quasi-Newton stochastic gradient algorithm by computing the Hessian of the log-likelihood; see Poyiadjis et al. (2005) for an application of this approach in a state-space model context.

To compute the gradient term appearing in (10), we note that

$$\begin{aligned}\nabla \log p_{\boldsymbol{\theta}}(z_t|z_{1:t-1}) &= \frac{\nabla p_{\boldsymbol{\theta}}(z_t|z_{1:t-1})}{p_{\boldsymbol{\theta}}(z_t|z_{1:t-1})} \\ &= \frac{\sum_{j=0}^{t-1} \nabla \xi_{\boldsymbol{\theta}}(C_t = j, z_{1:t})}{\sum_{j=0}^{t-1} \xi_{\boldsymbol{\theta}}(C_t = j, z_{1:t})}. \end{aligned} \tag{11}$$

By taking the derivative of $p_{\boldsymbol{\theta}}(C_t|z_{1:t})$ with respect to $\boldsymbol{\theta}$, we obtain

$$\nabla p_{\boldsymbol{\theta}}(C_t|z_{1:t}) = \frac{\nabla \xi_{\boldsymbol{\theta}}(C_t, z_{1:t})}{\sum_{i=0}^{t-1} \xi_{\boldsymbol{\theta}}(C_t = i, z_{1:t})} - p_{\boldsymbol{\theta}}(C_t|z_{1:t}) \frac{\sum_{i=0}^{t-1} \nabla \xi_{\boldsymbol{\theta}}(C_t = i, z_{1:t})}{\sum_{i=0}^{t-1} \xi_{\boldsymbol{\theta}}(C_t = i, z_{1:t})} \tag{12}$$

The term $\nabla \xi_{\boldsymbol{\theta}}(C_t, z_{1:t})$ is obtained by taking the derivative of (6)

$$\nabla \xi_{\boldsymbol{\theta}}(C_t = j, z_{1:t}) = \begin{cases} \begin{pmatrix} g_{\boldsymbol{\theta}}(z_t|C_t = j, z_{1:t-1}) f_{\boldsymbol{\theta}}(C_t = j|C_{t-1} = j) \\ \times p_{\boldsymbol{\theta}}(C_{t-1} = j|z_{1:t-1}) \pi_t^{(j,j)} \end{pmatrix} & \text{if } j < t-1 \\ \begin{pmatrix} g_{\boldsymbol{\theta}}(z_t|C_t = j, z_{1:t-1}) \\ \times \sum_{i=0}^{t-2} f_{\boldsymbol{\theta}}(C_t = j|C_{t-1} = i) p_{\boldsymbol{\theta}}(C_{t-1} = i|z_{1:t-1}) \pi_t^{(i,j)} \end{pmatrix} & \text{if } j = t-1 \end{cases} \tag{13}$$

where

$$\pi_t^{(i,j)} = \nabla \log g_{\boldsymbol{\theta}}(z_t|C_t = j, z_{1:t-1}) + \nabla \log f_{\boldsymbol{\theta}}(C_t = j|C_{t-1} = i) + \nabla \log p_{\boldsymbol{\theta}}(C_{t-1} = i|z_{1:t-1}).$$

4.2 Approximate inference

The computational cost of the recursion for computing $p_{\boldsymbol{\theta}}(C_t|z_{1:t})$ and $\nabla \log p_{\boldsymbol{\theta}}(z_t|z_{1:t-1})$ is proportional to t. This procedure is thus not appropriate for large datasets. We propose a deterministic approximation scheme to numerically approximate these quantities. Our approximation of $p_{\boldsymbol{\theta}}(C_t|z_{1:t})$ is inspired by the work of Fearnhead and Liu (2007) and relies on the following idea. At time t, the exact algorithm stores the set of probabilities $p_{\boldsymbol{\theta}}(C_t = j|z_{1:t})$ for $j = 0, 1, \ldots, t-1$. Given many of these probabilities are negligible, we can reasonably approximate the filtering distribution by a fewer set of N_t support points $c_t^{(1)}, \ldots, c_t^{(N_t)}$, called particles, with associated probability mass $w_t^{(1)}, \ldots, w_t^{(N_t)}$, called weights. To limit the number of

particles N_t at time t, we adopt a simple adaptive deterministic selection scheme where all the particles whose weights are below a given threshold ε are discarded; see below. In simulations, we have found that this deterministic selection step was performing better in terms of average mean square error than the random stratified optimal resampling proposed in Fearnhead and Liu (2007).

At time $t-1$, suppose that $\xi_{\boldsymbol{\theta}}(C_t, z_{1:t})$ and $p_{\boldsymbol{\theta}}(C_{t-1}|z_{1:t-1})$ are approximated through

$$\begin{aligned}\widehat{\xi}_{\boldsymbol{\theta}}(C_{t-1}, z_{1:t-1}) &= \sum_{i=1}^{N_{t-1}} \widetilde{w}_{t-1}^{(i)} \delta_{c_{t-1}^{(i)}}(C_{t-1}) \\ \widehat{p}_{\boldsymbol{\theta}}(C_{t-1}|z_{1:t-1}) &= \sum_{i=1}^{N_{t-1}} w_{t-1}^{(i)} \delta_{c_{t-1}^{(i)}}(C_{t-1}) \end{aligned} \tag{14}$$

where $\delta_{c_{t-1}^{(i)}}(C_{t-1}) = 1$ if $C_{t-1} = c_{t-1}^{(i)}$ and 0 otherwise. That is $\widetilde{w}_{t-1}^{(i)}$ resp. $w_{t-1}^{(i)}$ is an approximation of $\xi_{\boldsymbol{\theta}}\left(C_{t-1} = c_{t-1}^{(i)}, z_{1:t-1}\right)$ resp. $p_{\boldsymbol{\theta}}(C_{t-1} = c_{t-1}^{(i)} \Big| z_{1:t-1})$ and $w_{t-1}^{(i)} \propto \widetilde{w}_{t-1}^{(i)}$ with $\sum_{i=1}^{N_{t-1}} w_{t-1}^{(i)} = 1$. We propose to approximate $\nabla p_{\boldsymbol{\theta}}(C_{t-1}|z_{1:t-1})$ through

$$\widehat{\nabla p}_{\boldsymbol{\theta}}(C_{t-1}|z_{1:t-1}) = \sum_{i=1}^{N_{t-1}} w_{t-1}^{(i)} \beta_{t-1}^{(i)} \delta_{c_{t-1}^{(i)}}(C_{t-1}) \tag{15}$$

where $\sum_{i=1}^{N_{t-1}} w_{t-1}^{(i)} \beta_{t-1}^{(i)} = 0$; that is we are using the same particles $\left\{c_{t-1}^{(i)}\right\}$. Here $w_{t-1}^{(i)} \beta_{t-1}^{(i)}$ is an approximation of $\nabla p_{\boldsymbol{\theta}}(C_{t-1} = c_{t-1}^{(i)}|z_{1:t-1})$ so $\beta_{t-1}^{(i)}$ can be thought of as an approximation of $\nabla \log p_{\boldsymbol{\theta}}(C_{t-1} = c_{t-1}^{(i)}|z_{1:t-1})$.

At time t, let $\widetilde{c}_t^{(i)} = c_{t-1}^{(i)}$ and $\widetilde{c}_t^{(N_{t-1}+1)} = t-1$ for each particle $i = 1, \ldots, N_{t-1}$. To compute an approximation of $p_{\boldsymbol{\theta}}(C_{t-1}|z_{1:t-1})$, we plug our approximation (15) into (6) to obtain the unnormalized weights for $i = 1, \ldots, N_{t-1}$

$$\widetilde{w}_t^{(i)} = g_{\boldsymbol{\theta}}(z_t|C_t = \widetilde{c}_t^{(i)}, z_{1:t-1}) f_{\boldsymbol{\theta}}(C_t = \widetilde{c}_t^{(i)}|C_{t-1} = \widetilde{c}_t^{(i)}) w_{t-1}^{(i)}, \tag{16}$$

and

$$\widetilde{w}_t^{(N_{t-1}+1)} = g_{\boldsymbol{\theta}}(z_t|C_t = t-1, z_{1:t-1}) \sum_{i=0}^{N_{t-1}} f_{\boldsymbol{\theta}}(C_t = t-1|C_{t-1} = \widetilde{c}_t^{(i)}) w_{t-1}^{(i)}. \tag{17}$$

Similarly, by plugging (15) into (13), we obtain an approximation $\widetilde{\alpha}_t^{(i)}$ of $\nabla \xi_{\boldsymbol{\theta}}(C_t = \widetilde{c}_t^{(i)}, z_{1:t})$ which satisfies for $i = 1, \ldots, N_{t-1}$

$$\begin{aligned}\widetilde{\alpha}_t^{(i)} &= g_{\boldsymbol{\theta}}(z_t|C_t = \widetilde{c}_t^{(i)}, z_{1:t-1}) f_{\boldsymbol{\theta}}(C_t = \widetilde{c}_t^{(i)}|C_{t-1} = \widetilde{c}_t^{(i)}) w_{t-1}^{(i)} \\ &\times \left[\nabla \log g_{\boldsymbol{\theta}}(z_t|C_t = \widetilde{c}_t^{(i)}, z_{1:t-1}) + \nabla \log f_{\boldsymbol{\theta}}(C_t = \widetilde{c}_t^{(i)}|C_{t-1} = \widetilde{c}_t^{(i)}) + \beta_{t-1}^{(i)}\right],\end{aligned} \tag{18}$$

and

$$\begin{aligned}\widetilde{\alpha}_t^{(N_{t-1}+1)} &= g_{\boldsymbol{\theta}}(z_t|C_t = t-1, z_{1:t-1}) \sum_{i=0}^{N_{t-1}} f_{\boldsymbol{\theta}}(C_t = t-1|C_{t-1} = \widetilde{c}_t^{(i)}) w_{t-1}^{(i)} \\ &\times \left[\nabla \log g_{\boldsymbol{\theta}}(z_t|C_t = t-1, z_{1:t-1}) + \nabla \log f_{\boldsymbol{\theta}}(C_t = t-1|C_{t-1} = \widetilde{c}_t^{(i)}) + \beta_{t-1}^{(i)}\right].\end{aligned} \tag{19}$$

Using (11), we obtain

$$\widehat{\nabla \log p_{\boldsymbol{\theta}}}(z_t|z_{1:t-1}) = \frac{\sum_{i=1}^{N_{t-1}+1} \widetilde{\alpha}_t^{(i)}}{\sum_{i=1}^{N_{t-1}+1} \widetilde{w}_t^{(i)}}. \tag{20}$$

If we were to iterate this algorithm, the computational complexity would increase without bound with t. We only keep the particles $\widetilde{c}_t^{(i)}$ such that $\overline{w}_t^{(i)} > \varepsilon$ where $\overline{w}_t^{(i)} \propto \widetilde{w}_t^{(i)}$, $\sum_{i=1}^{N_{t-1}+1} \overline{w}_t^{(i)} = 1$ and discard the others. We then renormalize the weights of the surviving N_t particles and denote them $w_t^{(i)}$. Finally, using (12) we obtain

$$w_t^{(i)} \beta_t^{(i)} = \frac{\widetilde{\alpha}_t^{(\varphi(i))}}{\sum_{j=1}^{N_t} \widetilde{w}_t^{(\varphi(j))}} - w_t^{(i)} \frac{\sum_{j=1}^{N_t} \widetilde{\alpha}_t^{(\varphi(j))}}{\sum_{j=1}^{N_t} \widetilde{w}_t^{(\varphi(j))}} \tag{21}$$

for $i = 1, \ldots, N_t$ where $\varphi : \{1, \ldots, N_t\} \to \{1, \ldots, N_{t-1}+1\}$ is the injective function such that $w_t^{(i)} = \overline{w}_t^{(\varphi(i))}$.

To summarize, the particle filter for joint changepoints and parameter estimation proceeds as follows.

Particle filter for on-line changepoints and parameter estimation

At time $t = 1$

- Set $\boldsymbol{\theta}_0$, $c_1^{(1)} = 0$, $w_1^{(1)} = 1$, $w_1^{(1)} \beta_1^{(1)} = 0$ and $N_1 = 1$.

At time $t \geq 2$

- For $i = 1, \ldots, N_{t-1}$ let $\widetilde{c}_t^{(i)} = c_{t-1}^{(i)}$. Set $\widetilde{c}_t^{(N_{t-1}+1)} = t - 1$.
- For $i = 1, \ldots, N_{t-1} + 1$, compute $\widetilde{w}_t^{(i)}$ using (16-17) using $\boldsymbol{\theta}_{t-1}$.
- For $i = 1, \ldots, N_{t-1} + 1$, compute $\widetilde{\alpha}_t^{(i)}$ using (18-19) using $\boldsymbol{\theta}_{t-1}$.
- Update the parameter vector using (10) and (20), that is

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \gamma_t \frac{\sum_{i=1}^{N_{t-1}+1} \widetilde{\alpha}_t^{(i)}}{\sum_{i=1}^{N_{t-1}+1} \widetilde{w}_t^{(i)}}$$

- Use the adaptive selection step. Let N_t be the number of selected particles and $w_t^{(i)}$, $c_t^{(i)}$, $i = 1, \ldots, N_t$ be resp. the normalized weights and the associated support points and $\varphi : \{1, \ldots, N_t\} \to \{1, \ldots, N_{t-1} + 1\}$ the injective function such that $w_t^{(i)} = \widetilde{w}_t^{(\varphi(i))}$ for $i = 1, \ldots, N_t$.
- For $i = 1, \ldots, N_t$, compute the weights $w_t^{(i)} \beta_t^{(i)}$ using (21).

Note that a particle filter for joint state and parameter estimation relying on recursive maximum likelihood has also been proposed in Poyiadjis et al. (2005) for the class of general non-linear non-Gaussian state-space models. However, the cost of the algorithm in Poyiadjis et al. (2005) is quadratic in the number of particles whereas it is linear in our case.

Note that here T is the size of the dataset. The parameter estimate typically converges before time T for large T. For smaller datasets, we can run the particle filter $K > 1$ times on the dataset, using $\boldsymbol{\theta}_1^{(j)} = \boldsymbol{\theta}_T^{(j-1)}$ and $\gamma_1^{(j)} = \gamma_T^{(j-1)}$ as the initial values for parameter estimates and step size for runs $j = 2, \ldots, K$, so as to obtain convergence. In this case, the algorithm can be interpreted as a stochastic approximation algorithm maximizing $l_T(\boldsymbol{\theta})$ given by (8). Then the particle filter may be applied to obtain the MAP and full posterior of changepoints using the final hyperparameter estimate $\gamma_T^{(K-1)}$.

5 Experimental Data

5.1 Yeast dataset

We fitted our changepoint model to the positive strand of the first chromosome of the yeast data, using $u = 15$, $d = 2$, $\zeta_0 = 4$ and $\xi_0 = 10$, as explained earlier.

The parameters $\boldsymbol{\theta} = \{p_0, \psi_0, m_0, s_0, \nu_0, \gamma_0, \lambda, m_1, s_1, \nu_1, \gamma_1, \rho\}$ are estimated by running the particle filter $K = 20$ times with $\varepsilon = 10^{-6}$ on the full dataset, hence 4×10^5 iterations. Evolution of each parameter with respect to iterations is shown in Figure 1. Although we have used $K = 20$ passes over the whole dataset in order to show the convergence, most parameters had converged after only two passes. In terms of segmentation and classification, the results obtained using the parameter after 2 passes were very similar to the results obtained after 20 passes. Note that, as stated in the previous section, for a larger number of probes the parameter estimates would typically converge more quickly as there is more information. The final parameter value $\hat{\boldsymbol{\theta}}$ obtained after 20 passes over the full dataset, shown in Table 1, is used as the parameter values and the particle filter is then ran with $\varepsilon = 10^{-6}$ in order to obtain the segmentation. The MAP estimate of the changepoints for a portion of the whole chromosome is represented in Figure 2 (top). The associated number of changepoints for the whole chromosome is 299. Figure 2 (bottom) also shows the results using the algorithm of Huber et al. (2006), with 153 segments over chromosome 1. The number 153 was estimated using previous biological knowledge as explained in David et al. (2006). Overall, both segmentations show similar results and clearly agree with known coding sequence (CDS) annotations. This said, the advantage of our methodology over Huber's is obvious when looking at the segmentation results as we get a direct classification of the segments into transcripts and non-transcripts. Additionally,

no thresholding is necessary. Using our method, one can easily see that some of the detected transcripts (green background) do not overlap with know annotations. This confirms the findings of David et al. (2006) that even this well-studied genome has transcriptional complexity far beyond current annotations.

Note that, using our method, the number of segments is estimated whereas in Huber's it has to be fixed in advance. Our estimated number of segments is significantly larger than the number used by David et al. (2006), but a closer look at the segmentation results suggests that such a larger number is necessary to explain changes in intensity along the chromosome; see Figure 3 where we have zoomed onto two specific regions. For example, the left parts of Figure 3 (a) (around 6.9×10^4) show a clear jump in the observed intensities, which is detected as a separate transcript by our method (top) but not Huber's (bottom). Similar observations can be made for the left parts of Figure 3 (b) (around 1.14×10^5), where our method detects a putative transcript not detected by Huber's. Even though David et al. (2006) decided to fix the number of segments to 153 using previous biological knowledge, Huber et al. (2006) also provide a method for estimating the number of segments based on AIC or BIC. However, these require running the segmentation algorithm for all possible number of segments, which is not ideal for large genomes. For the data used here, the estimated number of segments using AIC and BIC are 307 and 232, respectively, which are closer to our estimate. Finally, Figure 3 (a) also shows the marginal posterior probabilities of changepoints, which provide nice measures of uncertainty for the corresponding changepoints. These marginal probabilities are obtained with 1,000 draws distributed from the approximated joint posterior distribution of the changepoints; see the algorithm in Section 4.1.1.

Overall, using the yeast data, we have shown that our changepoint model is a

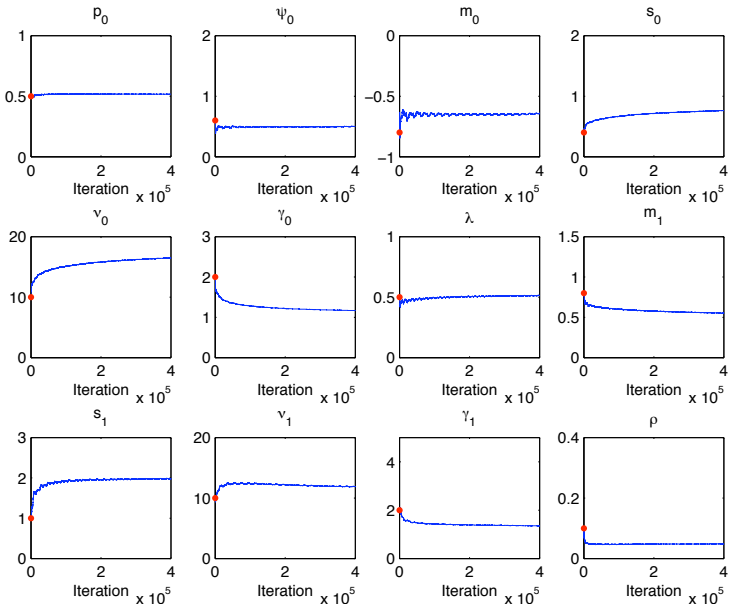


Figure 1: Online estimation convergence for the yeast data. The value of each parameter is shown as a function of iterations. The dot on the left-hand side of each plot represents the initial value of the hyperparameter.

Table 1: Summary of parameter estimates for both the yeast and human data, using the online estimation procedure.

Parameter	p_0	ψ_0	m_0	s_0	ν_0	γ_0	λ	m_1	s_1	ν_1	γ_1	ρ
Yeast	.52	.50	-0.64	.76	16.5	1.17	.51	.55	1.98	11.88	1.35	.05
Human	.23	.93	-0.37	.96	10.83	1.84	.63	1.18	.83	6.02	4.32	.21

compelling method for RNA transcript segmentation using tiling arrays as it automatically estimates the number of segments along with their classification while also estimating important tuning parameters. We now turn to a more complex human dataset (Cheng et al. 2005).

5.2 Human dataset

As with the yeast data, we fitted our changepoint model to the chromosome 6 of the human data with the same fixed parameters, namely $u = 15$, $d = 2$, $\zeta_0 = 4$ and $\xi_0 = 10$. For ease of comparison with Huber's segmentation algorithm, we have

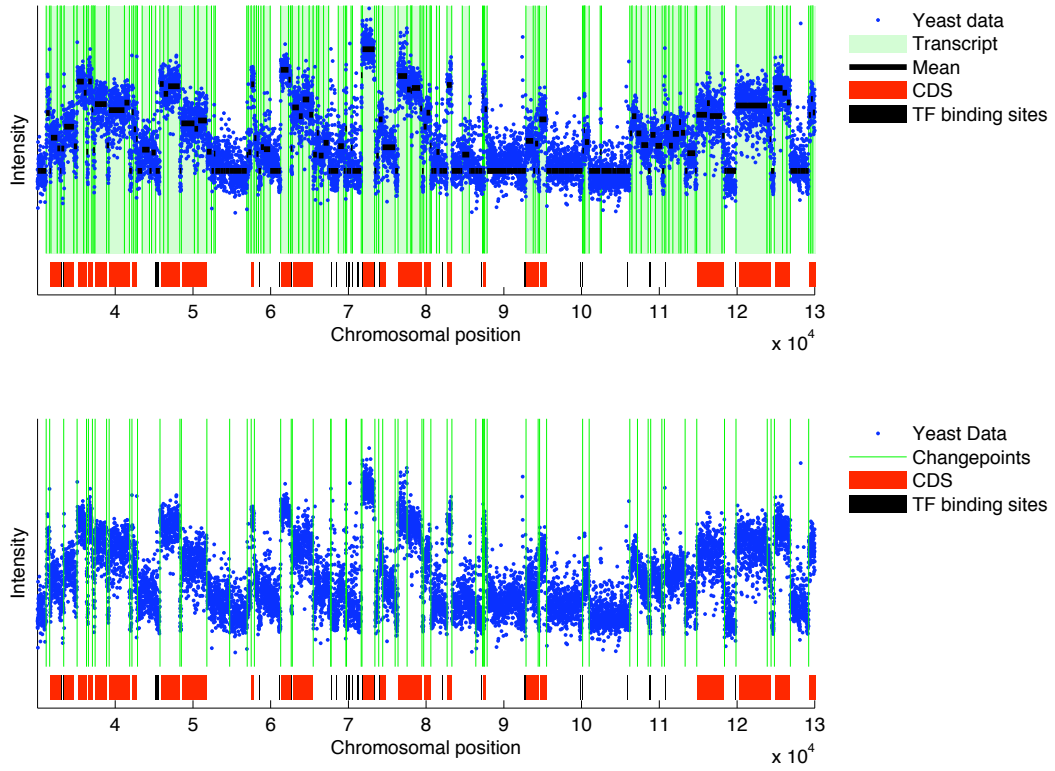
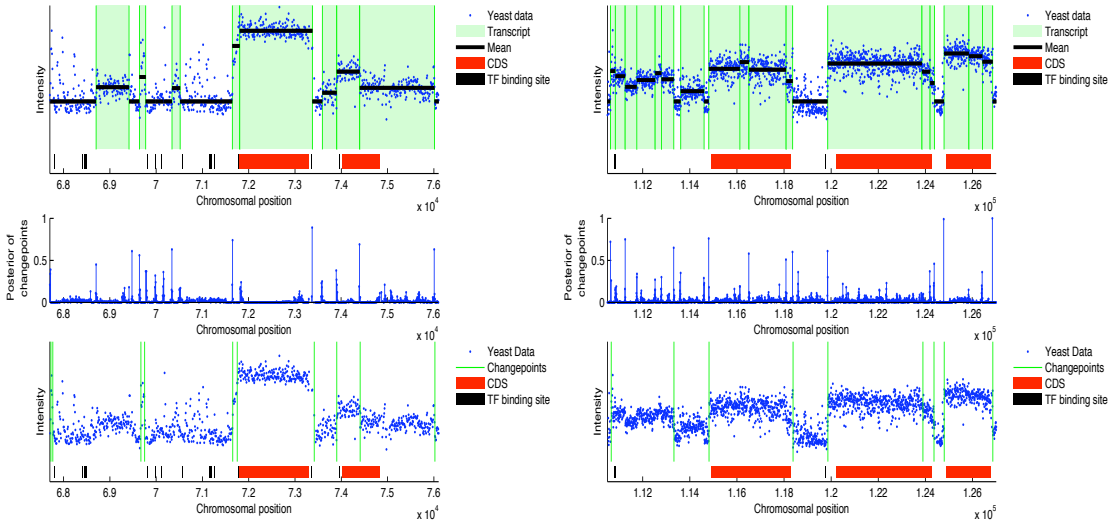


Figure 2: Segmentation results for part of chromosome 1 for the yeast data using our algorithm (top) and Huber et al.'s algorithm (bottom). For the top graph, the MAP estimate is displayed with transcript segments (green background), non-transcript segments (white background) segments and black segments for the segment intensity levels. For both top and bottom graphs, segments boundaries are represented with green vertical lines. Transcript annotations are shown below with red rectangles representing coding sequences and black segments representing TF binding sites.

only selected a subset of chromosome 6 which contains 20,000 probes with many known annotations and verified transcript regions. For comparison, we have also ran our algorithm on the whole chromosome 6, and the results were very similar. The parameters $\boldsymbol{\theta} = \{p_0, \psi_0, m_0, s_0, \nu_0, \gamma_0, \lambda, m_1, s_1, \nu_1, \gamma_1, \rho\}$ are first estimated, running the particle filter with $\varepsilon = 10^{-6}$ twenty times on the full dataset, hence 4×10^5 iterations. Evolution of $\hat{\boldsymbol{\theta}}$ with respect to iterations is shown in Figure 4. Most of the parameters have converged. The parameters associated to the skew t-



(a) (b)

Figure 3: Segmentation results for two close up regions from the yeast data. Our MAP segmentation (top) provides a better fit to the data by segmenting a few jumps in the data not detected as segments by Huber's method (bottom). The posterior probabilities of changepoints are represented in the middle plots.

distribution converge slowly due to the small probability of this mixture component (around 0.09). Using more iterations for the parameter estimation has shown very little difference for the changepoint results.

The parameter estimate $\hat{\boldsymbol{\theta}}$ obtained after 20 passes (Table 1), is used as the parameter value, then the particle filter algorithm is applied with $\varepsilon = 10^{-6}$ in order to obtain the segmentation. The MAP segmentation estimate of the changepoints for a portion of the whole chromosome is represented in Figure 5 (top). The associated number of changepoints is 824, which is significantly higher than for the yeast data used previously, even though it contains roughly the same number of probes. This is not surprising as the human genome is far more complex than its yeast counterpart as it contains many exons (Figure 5). In addition, the experiment of Cheng et al. (2005) was not strand-specific (hence the presence of both

+/- annotations on Figure 5) which could lead to more transcripts being detected. Finally, Cheng et al. (2005) did not use a control sample to normalize their data as did David et al. (2006), which could potentially lead to the detection of false transcripts due to sequence specific biases. Figure 5 (bottom) also shows the results using the algorithm of Huber et al. (2006), fixing the number of segments to 204 using BIC. Using AIC, the optimal number of segments is 5448, which seems a bit large, and requires running the algorithm several thousands times in order to select the optimal number of changepoints.

Because of the large number of changepoints, it is hard to compare our approach with that of Huber et al. (2006) based on Figure 5 alone. This said, the advantage of our methodology over Huber's is once again obvious when looking at the segmentation results as we get a direct classification of segments into transcripts and non-transcripts. In addition, using our method, the number of segments is estimated automatically. As with the yeast data, Figure 5 shows that many of the detected transcripts (green background) do not overlap with known annotations. This confirms the findings of Cheng et al. (2005), where the authors have noted that most of the detected transcripts were previously unannotated.

The number of segments estimated by our method is somewhat larger than the number estimated by Huber's segmentation combined with BIC, but a closer look at some specific regions suggests that such a number is necessary to explain changes in intensity along the chromosome; see Figure 6 where we have zoomed onto two specific regions. For example the left parts of Figure 6 (a) (around 7.1715×10^6) show many jumps in the observed intensities which are detected as separate transcripts by our method (top) but not Huber's (bottom). In fact, Huber's method fails to properly segment one validated region (mark as verified transcript). Figure 6 also shows the regions detected as transcripts by the sliding

window approach of Cheng et al. (2005). In general, our method and Huber's lead to precise estimates of the transcript boundaries whereas the sliding window approach tends to smooth out the boundaries, confirming previous observations made by Huber et al. (2006). In addition, the sliding window approach requires one to derive a threshold in order to call transcript regions, which can be difficult without prior knowledge. Cheng et al. (2005) used negative control measurements to derive the threshold used to detect transcripts, but such controls are not always available. Using our method, we simultaneously estimate the number of segments along with their classification (transcript/non-transcript). In particular, our method correctly classifies all of the verified transcripts. Note that such classification is not possible with Huber's method.

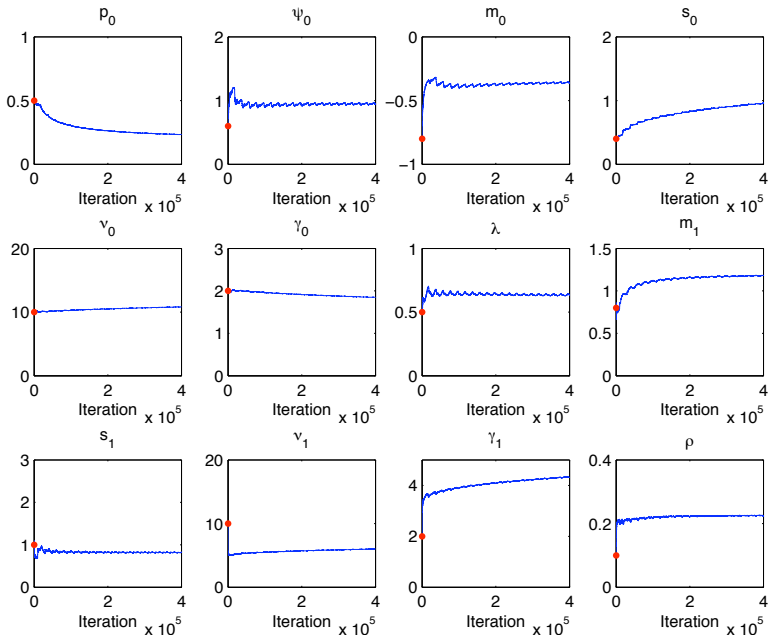


Figure 4: Parameter estimates for the human dataset. The value of each parameter is shown as a function of iterations. The dot on the left-hand side of each plot represents the initial parameter estimates.

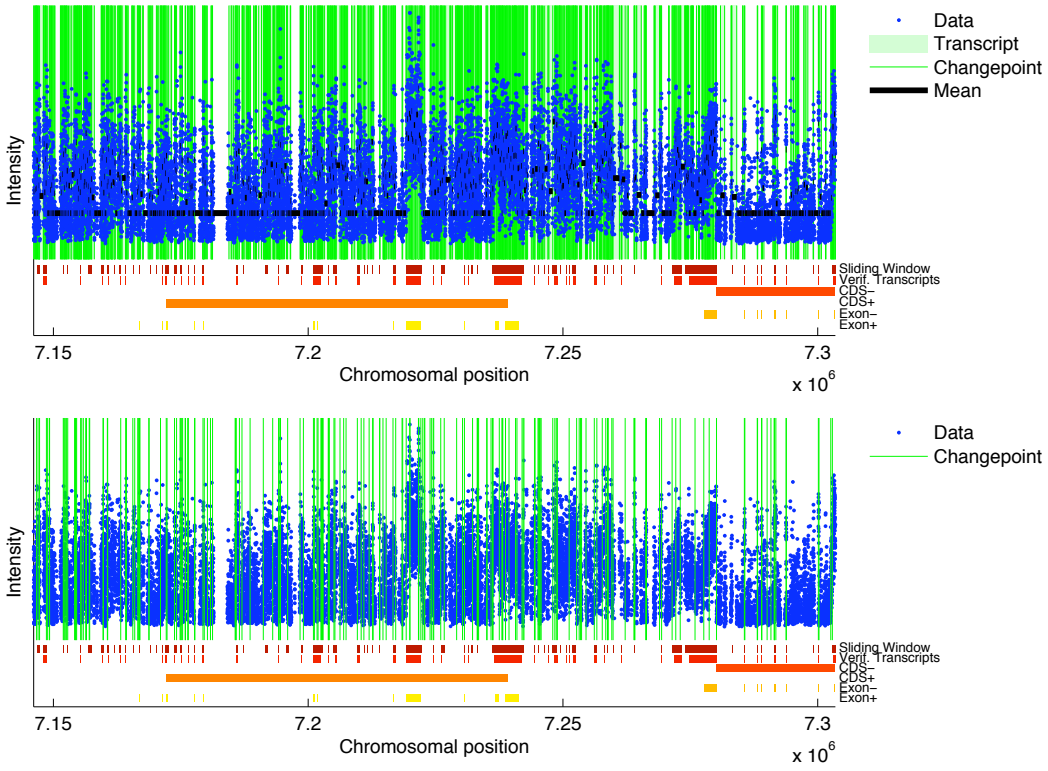


Figure 5: Segmentation results for part of chromosome 6 for the human dataset using our algorithm (top) and Huber et al.'s algorithm (bottom). For the top graph, the MAP estimate is displayed with transcript segments (green background) and non-transcript segments (white background), and black segments for the segment intensity levels. For both top and bottom graphs, segment boundaries are represented with green vertical lines. Transcript annotations are shown with coding sequences and Exon for both strands. We also show the transcript regions found by the sliding window method of Cheng et al. (2005), and the subset of these that were experimentally verified.

5.3 Model checking

In order to check model assumptions for both datasets, we now look at the predictive cumulative distribution $\Pr(Z_t \leq z_t | z_{1:t-1})$ evaluated at z_t. If the model assumptions are correct, these values should be uniformly distributed between 0 and 1 and $\Phi^{-1}(\Pr(Z_t \leq z_t | z_{1:t-1}))$, where Φ^{-1} is the inverse Gaussian cdf, should

be normally distributed. The histogram of the predictive distribution and the associated qq-plot are represented in Figure 7. Although the model is slightly overconfident, the histogram and qq-plot show that our model fits the data quite well for the yeast dataset. For the human dataset, the qq-plot and histogram are not as good, which is not surprising as the data are more noisy than the yeast data. Nonetheless, there is no evidence of severe mis-specification.

6 Simulated data

We have simulated a data set with 40,000 observations from our changepoint model described in Section 3 with the following parameters $p_0 = 0.4$, $\psi_0 = 0.47$, $\zeta_0 = 4$, $\xi_0 = 10$, $m_0 = -0.8$, $\varphi_0 = -1.27$, $s_0 = 0.3$, $\nu_0 = 16$, $\gamma_0 = 1.2$, $m_1 = 0.5$, $s_1 = 0.67$, $\nu_1 = 16$, $\gamma_1 = 1.2$, $\lambda = 0.35$, $\rho = 0.25$, $\alpha = 10^{-6}$, $d = 2$, $u = 15$. These values were chosen to be within the range of the estimated parameters in Table 1. The parameters $\boldsymbol{\theta}$ are first estimated on the whole dataset. The evolution of the parameter estimates over time are represented in Figure 8. The algorithm manages to correctly estimate this set of parameters. Based on the final estimated value, the particle filter is then run again on the whole dataset. The MAP, posterior of changepoints, and number of particles for a portion of the data are represented in Figure 9. The true transcribed segments are represented by red patches. The number of particles varies over time adaptively. It increases as long as there is no changepoint, and decreases when evidence of a changepoint occurs. The predictive histogram and qq-plot are given in Figure 10. Even with a few number of particles (20 on average), the algorithm manages to estimate the model parameters (including the segment boundaries) very well.

7 Discussion

We have developed a flexible changepoint model combined with an on-line parameter estimation method which provides a powerful framework for detecting RNA transcripts from tiling array experiments. Our method presents several advantages over current approaches. In particular, it does not suffer from degeneracy problems of standard particle methods for static parameter estimation (Andrieu et al. 2004; Fearnhead 2002). Additionally, it can automatically detect the number of segments and call transcript regions. In addition, the estimation algorithm is linear in the number of features, which is an important characteristic for whole genome analysis where the number of features is very large. Using two experiments on Affymetrix tiling arrays, we have shown that our approach can provide powerful detection of RNA transcript compared to a sliding window approach or a simple segmentation algorithm. This is particularly true of the human dataset were we have detected all of the verified transcripts. In addition, we have performed a simulation study which showed that our estimation procedure provides good estimates of the unknown parameters, including the unknown changepoints.

Here we have assumed that the biological process of transcription can be described by piecewise constant expression levels. In reality, the actual biological process could lead to more complex hybridization profiles than the piecewise constant shape assumed here. In addition, we have assumed that conditioning on the changepoints, the residuals are independent. In reality, the residuals might not be independent due to complex biological processes and overlaps in probe sequences. This said, we view our model as a useful approximation to the true biological process that can be used to detect meaningful transcripts as demonstrated with the two experimental datasets explored here.

Finally, even though our model was developed for RNA transcript analysis, the methodology introduced is far more general and could be used in many other problems (e.g. copy number variation), other type of arrays (e.g. Nimblegen, Agilent), and also the applications discussed in Fearnhead and Liu (2007).

Acknowledgment

The authors are grateful to Luke Bornn for helpful comments and Philipp Kapranov for helpful discussion about the human data. The first author thanks DGA for its support.

References

Andrieu, C., A. Doucet, S. Singh, and V. Tadic (2004). Particle methods for change detection, identification and control. *Proceedings of the IEEE 92*, 423–438.

Azzalini, A. and A. Capitanio (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution. *Journal of the Royal Statistical Society B 65*, 367–389.

Barry, D. and J. Hartigan (1992). Product partition models for change point problems. *The Annals of Statistics 20*, 260–279.

Benveniste, A., M. Metivier, and P. Priouret (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag.

Bertone, P., V. Stolc, T. E. Royce, J. S. Rozowsky, A. E. Urban, X. Zhu, J. L. Rinn, W. Tongprasit, M. Samanta, S. Weissman, M. B. Gerstein, and M. Snyder (2004). Global identification of human transcribed sequences with genome tiling arrays. *Science 306*(5705), 2242–6.

Bolstad, B. M., R. A. Irizarry, M. Astrand, and T. P. Speed (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics 19*(2), 185–93.

Carroll, J. S., X. S. Liu, A. S. Brodsky, W. Li, C. A. Meyer, A. J. Szary, J. Eeckhoute, W. Shao, E. V. Hestermann, T. R. Geistlinger, E. A. Fox, P. A. Silver, and M. Brown (2005). Chromosome-wide mapping of estrogen receptor binding reveals long-range regulation requiring the forkhead protein foxa1. *Cell 122*(1), 33–43.

Cheng, J., P. Kapranov, J. Drenkow, S. Dike, S. Brubaker, S. Patel, J. Long, D. Stern, H. Tammana, G. Helt, V. Sementchenko, A. Piccolboni, S. Bekiranov, D. K. Bailey, M. Ganesh, S. Ghosh, I. Bell, D. S. Gerhard, and T. R. Gingeras (2005). Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science 308*(5725), 1149–54.

Chopin, N. (2007). Dynamic detection of change points in long time series. *Annals of the Institute of Mathematical Sciences 59*, 349–366.

David, L., W. Huber, M. Granovskaia, J. Toedling, C. J. Palm, L. Bofkin, T. Jones, R. W. Davis, and L. M. Steinmetz (2006). A high-resolution map of transcription in the yeast genome. *Proc Natl Acad Sci USA 103*(14), 5320–5.

Do, K., P. Muller, and F. Tang (2005). A bayesian mixture model for differential gene expression. *Journal of The Royal Statistical Society C 54*, 627–644.

Efron, B. (2004). Large-scale simultaneous hypothesis testing: The choice of a null hypothesis. *Journal of The American Statistical Association 99*(465), 96–104.

Fearnhead, P. (2002). MCMC, sufficient statistics and particle filter. *Journal of*

Computational and Graphical Statistics 11, 848–862.

Fearnhead, P. and Z. Liu (2007). On-line inference for multiple change points problems. *Journal of the Royal Statistical Society B 69*, 589–605.

Gentleman, R. C., V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. A. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol 5*(10), R80.

Gottardo, R., J. A. Pannucci, C. R. Kuske, and T. S. Brettin (2003). Statistical analysis of microarray data: a bayesian approach. *Biostatistics 4*(4), 597–620.

Gottardo, R., A. E. Raftery, K. Y. Yeung, and R. E. Bumgarner (2006). Bayesian robust inference for differential gene expression in microarrays with multiple samples. *Biometrics 62*(1), 10–8.

Huber, W., J. Toedling, and L. M. Steinetz (2006). Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics 22*(16), 1963–1970.

Irizarry, R. A., B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics 4*(2), 249–64.

Kapranov, P., S. E. Cawley, J. Drenkow, S. Bekiranov, R. L. Strausberg, S. P. A. Fodor, and T. R. Gingeras (2002). Large-scale transcriptional activity in chromosomes 21 and 22. *Science 296*(5569), 916–9.

Lockhart, D. J., H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol 14*(13), 1675–80.

Mockler, T. C., S. Chan, A. Sundaresan, H. Chen, S. E. Jacobsen, and J. R. Ecker (2005). Applications of dna tiling arrays for whole-genome analysis. *Genomics 85*(1), 1–15.

Newton, M. A., C. M. Kendziorski, C. S. Richmond, F. R. Blattner, and K. W. Tsui (2001). On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *J Comput Biol 8*(1), 37–52.

Poyiadjis, G., A. Doucet, and S. S. Singh (2005). Maximum likelihood parameter estimation in general state-space models using particle methods. In *Joint Statistical Meeting*, Minneapolis.

A Distributions

The probability distribution of the skew t-distribution with parameters φ_0, ψ_0, ζ_0 and ξ_0 is given by

$$\frac{2}{\psi_0} t\left(\frac{x-\varphi_0}{\psi_0}, \zeta_0\right) T\left(\xi_0\left(\frac{x-\varphi_0}{\psi_0}\right)\sqrt{\frac{\zeta_0+1}{\left(\frac{x-\varphi_0}{\psi_0}\right)^2+\zeta_0}}, \zeta_0+1\right) \tag{22}$$

where t and T are the standard centered student t density and cumulative density function, respectively. The parameters φ_0, ψ_0, ζ_0 and ξ_0 represent the location, scale, degrees of freedom and skewness parameters. The normal inverse gamma distribution $(\mu, \sigma^2) \sim \mathcal{N}i\mathcal{G}(m_1, s_1, \nu_1, \gamma_1)$ is defined by

$$(\mu_i|\sigma_i^2) \sim \mathcal{N}(m_1, s_1^2\sigma_i^2), \qquad \sigma_i^2 \sim i\mathcal{G}(\frac{\nu_1}{2}, \frac{\gamma_1}{2})$$

and the resulting joint pdf is given by

$$\mathcal{N}i\mathcal{G}(\mu_i, \sigma_i^2|m_1, s_1, \nu_1, \gamma_1) = (2\pi s_1^2\sigma_i^2)^{-1/2}\exp(-\frac{1}{2s_1^2\sigma_i^2}(x-m_1)^2) \\ \times \frac{(\frac{\gamma_1}{2})^{\frac{\nu_1}{2}}}{\Gamma(\frac{\nu_1}{2})}(\sigma_i^2)^{-\frac{\nu}{2}-1}\exp(-\frac{\gamma_1}{2\sigma_i^2}).$$

B Marginal likelihoods

B.1 Transcribed segments

The marginal likelihood $p(z_{\tau_i+1:\tau_{i+1}}|r_i = 1)$ is

$$\begin{aligned} p(z_{\tau_i+1:\tau_{i+1}}|r_i = 1) &= \int p(z_{\tau_i+1:\tau_{i+1}}|r_i = 1, \mu_i, \sigma_i^2)p(\mu_i, \sigma_i^2)d\mu_i d\sigma_i \\ &= \pi^{-n/2}(1+ns_1^2)^{-1/2}(s^2 + \frac{n(m-m_1)^2}{1+ns_1^2} + \gamma_1)^{-(n+\nu_1)/2} \\ &\times \gamma_1^{\nu_1/2}\frac{\Gamma((n+\nu_1)/2)}{\Gamma(\nu_1/2)} \end{aligned}$$

where $m = \frac{1}{n}\sum_{k=\tau_i+1}^{\tau_{i+1}} z_k$, $s^2 = \sum_{k=\tau_i+1}^{\tau_{i+1}}(z_k - m)^2$ and $n = \tau_{i+1} - \tau_i$.

B.2 Non-transcribed segments

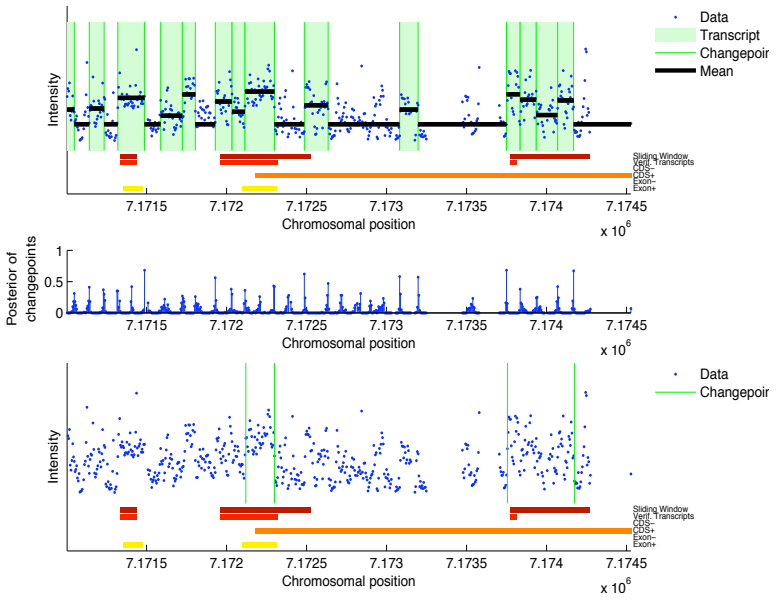
The marginal likelihood $p(z_{\tau_i+1:\tau_{i+1}}|r_i = 0)$ is

$$p(z_{\tau_i+1:\tau_{i+1}}|r_i = 0) = (1-p_0)p(z_{\tau_i+1:\tau_{i+1}}|r_i = 0, q_i = 0) + p_0 p(z_{\tau_i+1:\tau_{i+1}}|r_i = 0, q_i = 1)$$

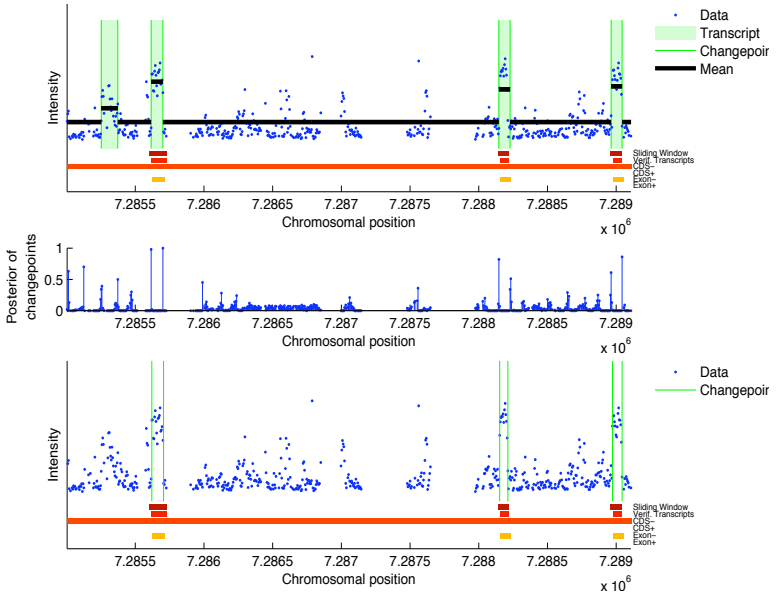
where

$$\begin{aligned} p(z_{\tau_i+1:\tau_{i+1}}|r_i = 0, q_i = 0) &= \prod_{k=\tau_i+1}^{\tau_{i+1}} st(z_k; \varphi_0, \psi_0, \zeta_0, \xi_0) \\ p(z_{\tau_i+1:\tau_{i+1}}|r_i = 0, q_i = 1) &= \pi^{-n/2}(1+ns_0^2)^{-1/2}\left(s^2 + \frac{n(m-m_0)^2}{1+ns_0^2} + \gamma_0\right)^{-(n+\nu_0)/2} \\ &\times \gamma_0^{\nu_0/2}\frac{\Gamma((n+\nu_0)/2)}{\Gamma(\nu_0/2)} \end{aligned}$$

where again $m = \frac{1}{n}\sum_{k=\tau_i+1}^{\tau_{i+1}} z_k$, $s^2 = \sum_{k=\tau_i+1}^{\tau_{i+1}}(z_k - m)^2$ and $n = \tau_{i+1} - \tau_i$.



(a)



(b)

Figure 6: Segmentation results for two close-up regions from the human data. Our MAP segmentation (top) provides a better fit to the data and properly detect a verified transcript not detected by Huber's method (bottom).

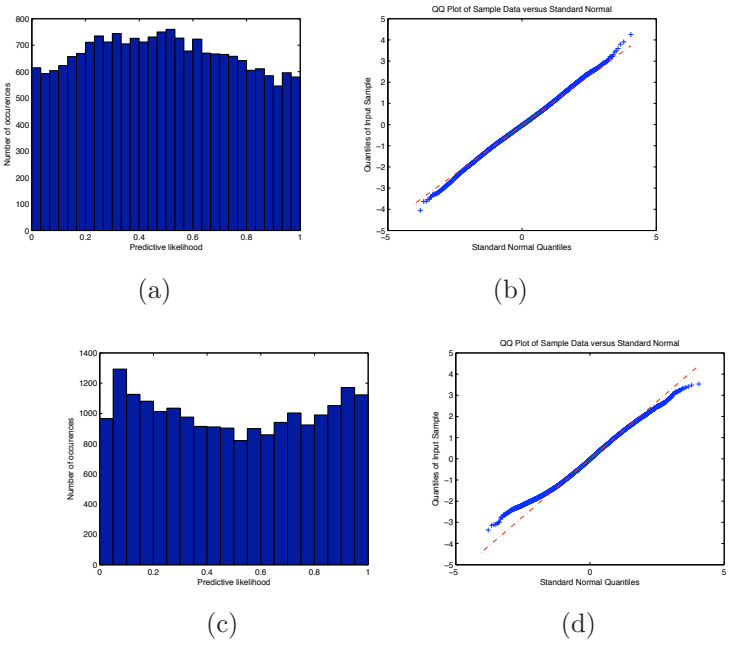


Figure 7: Histogram of $\Pr(Z_t \leq z_t | z_{1:t-1})$ and qq-plot for the Yeast (a-b) and Human (c-d) Datasets.

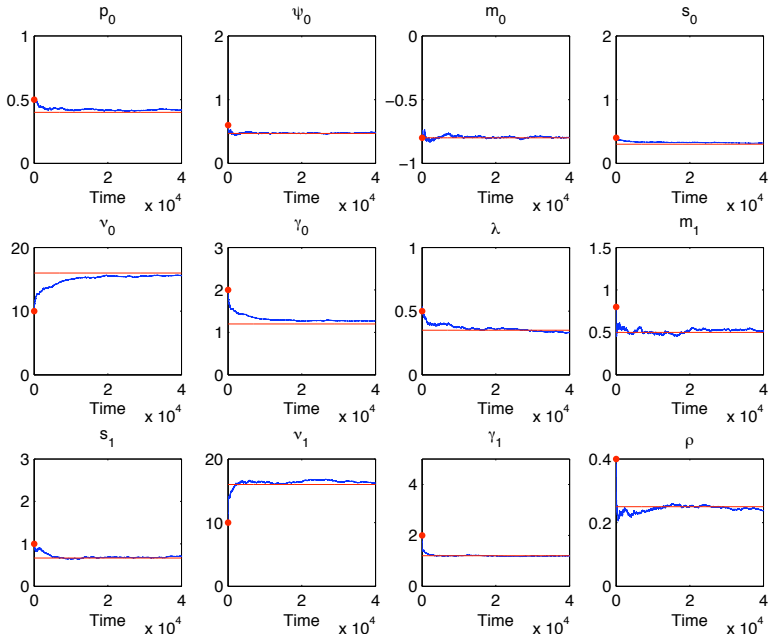


Figure 8: Parameter estimates for the simulated data. The value of each parameter is shown as a function of iterations. The true value for each parameter is shown with a horizontal line.

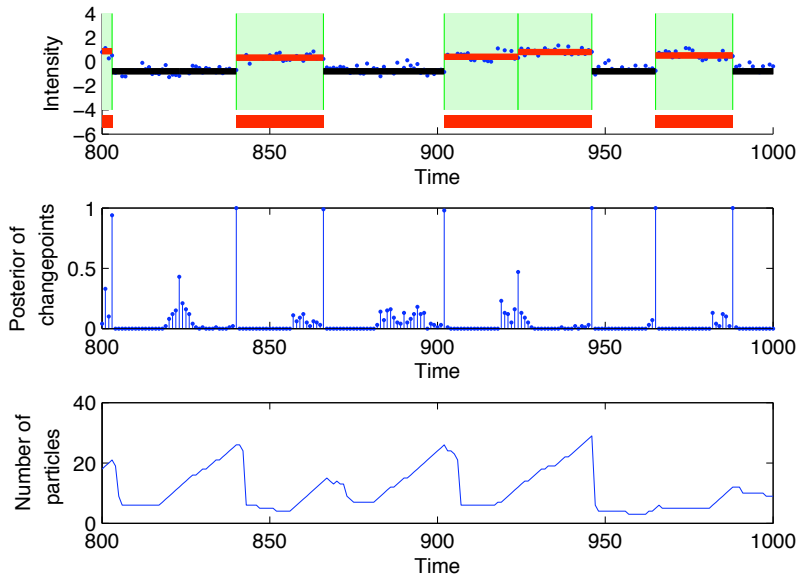
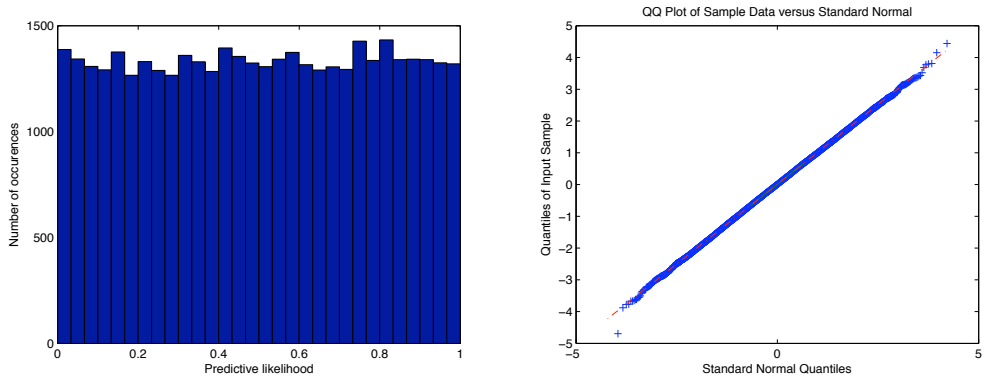


Figure 9: (top) MAP, (middle) posterior of changepoints and (bottom) number of particles for the simulated dataset. On the top figure, the true transcribed segments are represented by red patches.



(a) (b)

Figure 10: Histogram of $\Pr(Z_t \leq z_t | z_{1:t-1})$ and qq-plot for the simulated dataset