Supplementary Information for Parseq: reconstruction of microbial transcription landscape from RNA-Seq read counts using state-space models *(submitted)*

Bogdan Mirauta[1,⋆], Pierre Nicolas[2,†] and Hugues Richard[1,†]

[1] Génomique des Microorganismes, UPMC and CNRS UMR7238, Paris, France.

[2] Mathématique Informatique et Génome, INRA UR1077, Jouy-en-Josas, France.

⋆ To whom correspondence should be addressed. † Contributed equally to this work.

November 30, 2013

Contents

1 The Parseq model for RNA-seq data

1.1 Directed Acyclic Graph

Directed Acyclic Graphs (DAGs) representing the conditional dependence between the variables (i.e. the factorization of the likelihood) provide a convenient way to summarize models involving multiple layers of variables and to understand conditional independence relationships. The DAG of our model is shown Figure S1. Of note, transition kernels for $\mathbf{u}$ and $\mathbf{s}$ as well as the emission model for $\mathbf{y}$ write as mixtures of several types of distributions. In line with the classical data augmentation strategy, auxiliary variables ($\mathbf{o}$, $\mathbf{bk_u}$, and $\mathbf{bk_s}$) have been added to disambiguate the contribution of the different components of each of these mixtures in order to facilitate the design of the MCMC.

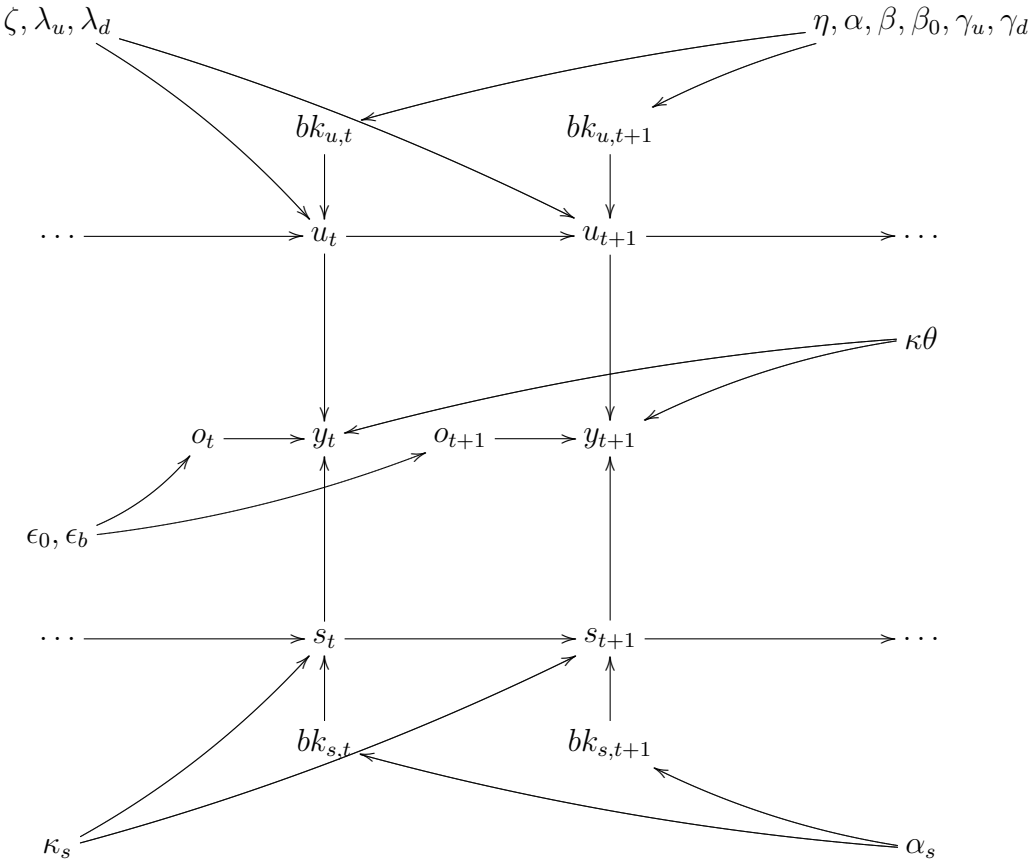

Figure S1 The DAG of the Parseq model for RNA-seq data along the genome. The main variables are: $\mathbf{y} = (y_t)_{t\geq 1}$, the sequence of observations (read counts); $\mathbf{u} = (u_t)_{t\geq 1}$, the underlying expression level that we try to reconstruct, and $\mathbf{s} = (s_t)_{t\geq 1}$, the local scaling variable that accounts for correlated overdispersion of read counts. Three auxiliary variables have been added to make easier the design of the MCMC algorithm: $\mathbf{o} = (o_t)_{t\geq 1}$ indicating whether the observation has been generated by the outlier model or not; $\mathbf{bk_u} = (bk_{u,t})_{t\geq 1}$ indicating which type of transition occurred between adjacent positions on $\mathbf{u}$ (shift, drift, no-change); $\mathbf{bk_s} = (bk_{s,t})_{t\geq 1}$ indicating which type of transition occurred between adjacent positions on $\mathbf{s}$ (change vs. no-change). Parameters are denoted by Greek letters and for clarity inserted in the DAG by curved arrows. The variables that contribute to the emission model but are marginalized out in the final formula (a_t, x_t), are not shown here.

1.2 Notations

Throughout this document we use the following probability distributions: $\mathcal{P}(\lambda)$ is the Poisson distribution parametrized by mean λ (and $\mathcal{P}(x;\lambda)$ denotes to the corresponding density at point x); $\mathcal{NB}(r,p)$ is the Negative Binomial distribution with size r and probability p (mean $\frac{rp}{1-p}$); $\mathcal{U}(a,b)$ is the uniform distribution between a and b; $\Gamma(\kappa,\theta)$ is the Gamma distribution with shape κ and scale θ; $\mathcal{E}(\lambda)$ is the exponential distribution with rate λ; $\mathbf{1}_c$ is the indicator function whose value is 1 if condition c is true and 0 otherwise; δ_d is the Dirac delta function with unit mass at point d.

1.3 Markov transition kernel for the expression level u

The Markov transition kernel for the expression level **u** writes

$$\begin{aligned} k_u(u_{t+1};u_t) &= \mathbf{1}_{\{u_t=0\}}\cdot\left[(1-\eta)\delta_0(u_{t+1})+\eta\cdot\zeta\cdot e^{-u_{t+1}\cdot\zeta}\right] \\ &+ \mathbf{1}_{\{u_t>0\}}\cdot\Big[\alpha\cdot\delta_{u_t}(u_{t+1})+\beta\cdot\zeta\cdot e^{-u_{t+1}\cdot\zeta}+\beta_0\cdot\delta_0(u_{t+1}) \\ &\quad +\gamma_u\mathbf{1}_{\{u_{t+1}>u_t\}}\cdot\frac{\lambda_u}{u_t}\cdot e^{-\frac{\lambda_u\cdot(u_{t+1}-u_t)}{u_t}}+\gamma_d\cdot\mathbf{1}_{\{u_{t+1}<u_t\}}\cdot\frac{u_t}{u_{t+1}}\cdot\frac{\lambda_d}{u_{t+1}}\cdot e^{-\frac{\lambda_d\cdot(u_t-u_{t+1})}{u_{t+1}}}\Big], \end{aligned}$$

The Markov transition kernel between genome positions t and $t+1$ is best understood as a mixture of seven types of moves that are allowed or not depending on whether position t is in an expressed region ($u_t>0$) or not ($u_t=0$), as described below. The type of the move between positions t and $t+1$ is recorded in the auxilliary variable $bk_{u,t+1}$.

- When t is in a non-expressed region ($u_t=0$):
  - An expressed region starts with probability η. In this case $bk_{u,t+1}$ = 'shift out of 0' and the density of u_{t+1} is drawn from an exponentional distribution with rate ζ (density $\zeta\cdot e^{-u_{t+1}\cdot\zeta}$).
  - The non-expressed region continues with $1-\eta$. In this case $bk_{u,t+1}$ = 'no-change in 0' and distribution of u_{t+1} is a point mass at $u_t=0$ (density $\delta_0(u_{t+1})$).
- When t is in an expressed region ($u_t>0$):
  - The expression level remains unchanged with probability α. In this case $bk_{u,t+1}$ = 'no-change' and the distribution of u_{t+1} is a point mass at $u_t>0$ (density $\delta_{u_t}(u_{t+1})$).
  - The expression level exhibits a small amplitude change (drift) upward with probability γ_u. In tis case $bk_{u,t+1}$ = 'upward drift' and the distribution of u_{t+1} is such that u_{t+1}/u_t is drawn from an exponential distribution with rate λ_u (density of u_{t+1} is $(\lambda_u/u_t)\cdot e^{-\lambda_u\cdot(u_{t+1}-u_t)/u_t}$).
  - The expression level exhibits a small amplitude change (drift) downward with probability γ_d. In tis case $bk_{u,t+1}$ = 'downward drift' and the distribution of u_{t+1} is such that u_t/u_{t+1} is drawn from an exponential distribution with rate λ_d (density of u_{t+1} is $(u_t/u_{t+1})\cdot(\lambda_d/u_{t+1})\cdot e^{-\lambda_d\cdot(u_t-u_{t+1})/u_{t+1}}$).
  - The expression level changes to a value independent of the previous position with probability β. In this case $bk_{u,t+1}$ = 'shift' and the density of u_{t+1} is drawn from an exponentional distribution with rate ζ (density $\zeta\cdot e^{-u_{t+1}\cdot\zeta}$).

– The expressed region ends with probability β_0. In this case $bk_{u,t+1}$ = 'shift to 0' and the distribution of u_{t+1} is a point mass at 0 (density $\delta_0(u_{t+1})$).

The probabilities of the five different types of moves allowed when $u_t > 0$ sum to 1, *i.e.* $\alpha + \gamma_u + \gamma_d + \beta + \beta_0 = 1$.

1.4 Markov transition kernel for the local scaling term s

The Markov transition kernel for the local scaling term **s** writes

$$k_s(s_{t+1}; s_t) \;=\; \alpha_s \cdot \delta_{s_t}(s_{t+1}) + (1-\alpha_s) \cdot \Gamma(s_{t+1}; shape = \kappa_s, scale = 1/\kappa_s)\,.$$

This corresponds to a piecewise constant Gamma distributed value with mean 1 and standard deviation $1/\sqrt{\kappa_s}$ (shape κ_s and scale $1/\kappa_s$). The probability of change between t and $t+1$ is α_s and the type of move ('move' or 'no-move') is recorded in $bk_{s,t+1}$.

1.5 Read count emission model

The variable y_t that we model here corresponds to the number of reads with 5'-ends mapping at position t.
Given the number of molecules initially sampled x_t, and the amplification coefficient a_t, the probability density function of the observed read count writes

$$e(y_t; x_t, a_t) \;=\; (1-\varepsilon_b-\varepsilon_o) \cdot \mathcal{P}(y_t; x_t \cdot a_t) + \varepsilon_b \cdot \mathcal{P}_{-\{0\}}(y_t; 1 \cdot a_t) + \varepsilon_o \cdot \mathcal{U}(y_t; 0 \ldots b)\,.$$

Using the distributions of the number of molecules and of the amplification ($x_t \sim \mathcal{P}(\frac{u_t \cdot s_t}{\kappa\theta})$, $a_t \sim \Gamma(\kappa, \theta)$) we rewrite the emission in terms of expression level u_t and local scaling s_t as

$$\begin{aligned} e(y_t; u_t, s_t) \;&=\; (1-\varepsilon_b-\varepsilon_o) \cdot \sum_{x_t=0}^{\infty} \mathcal{P}(x_t; \frac{u_t s_t}{\kappa\theta}) \cdot \mathcal{NB}(y_t; \kappa, \frac{x_t\theta}{x_t\theta+1}) \\ &+\; \varepsilon_b \cdot \mathcal{NB}_{-\{0\}}(y_t; \kappa, \frac{\theta}{\theta+1}) + \varepsilon_o \cdot \mathcal{U}(y_t; 0 \ldots b)\,. \end{aligned}$$

This corresponds to a mixture of three components whose type is recorded in variable o_t.

- with probability $1-\varepsilon_b-\varepsilon_o$ the observed read counts depends on the underlying transcription level whose distribution writes itself as a mixture

$$y_t \sim \sum_{x_t=0}^{\infty} \mathcal{P}(x_t; \frac{u_t s_t}{\kappa\theta}) \cdot \mathcal{NB}(\kappa, \frac{x_t\theta}{x_t\theta+1})\,.$$

  In this case o_t = 'normal'.

- with probability ε_b the observed read counts come from the background noise, whose distribution is shaped by amplification but the initial of number of molecules is assumed to be small (this corresponds to $x_t = 1$ given $y_t \geq 1$). The distribution of these counts arising from background noise writes thus as the zero-truncated NB distribution $\mathcal{NB}_{-\{0\}}(\kappa, \frac{\theta}{\theta+1})$.In this case o_t = 'background'.

- with probability ε_o the observed read counts come from an uniform distribution between 0 and b (in practice b is set to $\max(\mathbf{y})$). In this case o_t = 'outlier'.

2 Reconstruction of u and s via SMC

2.1 Particle Gibbs with block update

Sequential Monte Carlo (SMC) approaches are used in our Markov Chain Monte Carlo algorithm to build updates of the hidden trajectories **u** and **s** that preserve their conditional distributions. We initially tested the use of a traditional forward filtering approach (Doucet and Johansen, 2009) as an approximate solution to sample the conditional distribution but we latter favored the use of a recently described algorithm 'Particle Gibbs' (Andrieu *et al.*, 2010) that provides an exact solution to this problem. The central idea of the Particle Gibbs algorithm is to rely on a new type of SMC, termed Conditional SMC, that lead to accept or not the proposed changes of the trajectory with an appropriate probability to ensure exactness (this can be thought as a Metropolis-Hastings algorithm where the proposal is provided by SMC).

In our context an additional difficulty comes from the length of the sequence which leads to very small acceptance probability and thus poor mixing for any reasonable number of particles. We implemented a block update version of the Particle Gibbs algorithm to circumvent this problem. Of note, our piecewise constant models for u and s impose restriction on the selection of the blocks to ensure reversibility of the Markov chain generated by the Particle Gibbs MCMC. The procedure to select the blocks is explained after the description of the block update by Conditional SMC.

The block update Conditional SMC algorithm for **u** between positions t_0 and $t_1 > t_0$ ($u_{t_0:t_1}$) writes

1. For $t = t_0$,
   (a) For the first $P-1$ particles $1 \leq p < P$, draw $\tilde{u}_t^p$ from the proposal density $q_{u,t_0}(\tilde{u}_{t_0}^p; u_{t_0-1})$ where q_{u,t_0} is a position-specific proposal that uses the observed read-counts to focus sampling near the target (see subsection 2.2).
   (b) Set $\tilde{u}_t^N = u_t$
   (c) For $1 \leq p \leq P$, calculate the normalized particle weights $w_{t_0}^p \propto \frac{k_u(\tilde{u}_{t_0}^p; u_{t_0-1}) \cdot e(y_{t_0}; \tilde{u}_{t_0}^p, s_{t_0})}{q_{u,t_0}(\tilde{u}_{t_0}^p; u_{t_0-1})}$ such as $\sum_{p=1:P} w_{t_0}^p = 1$
2. From $t = t_0 + 1$ to $t = t_1$,
   (a) If $1/\sum_{p=1}^{P}(w_{t-1}^p)^2 > P/4$ (the effective sampling size is above $P/4$) then $a_t^p = p$ for $1 \leq p \leq P$ and $w_t'^p = w_{t-1}^p$.
   (b) If $1/\sum_{p=1}^{P}(w_{t-1}^p)^2 \leq P/4$ then the index of the ancestor particle a_t^p is drawn from a multinomial distribution with weights $(w_{t-1}^p)_{1:P}$ for $1 \leq p < P$, $a_t^P = P$ and $w_t'^p = 1$ for $1 \leq p \leq P$.
   (c) For $1 \leq p < P$, draw $\tilde{u}_t^p$ from the proposal density $q_{u,t}(\tilde{u}_t^p; \tilde{u}_{t-1}^{a_t^p})$.
   (d) Set $\tilde{u}_t^P = u_t$.
   (e) For $1 \leq p \leq P$, calculate the normalized particle weights $w_t^p \propto w_t'^p \cdot \frac{k_u(\tilde{u}_t^p; \tilde{u}_{t-1}^{a_t^p}) \cdot e(y_t; \tilde{u}_t^p, s_t)}{q_{u,t}(\tilde{u}_t^p; \tilde{u}_{t-1}^{a_t^p})}$ such as $\sum_{p=1:P} w_t^p = 1$
3. For $t = t_1$,

(a) Sample the index $p^\star_{t_1}$ of particle corresponding to the new trajectory at position t_1. Each particle p has a probability proportional to $w^p_{t_1} \cdot k_u(u_{t_1+1}; \tilde{u}^p_{t_1})$.

(b) Backtrace the whole trajectory from $t = t_1$ to $t = t_0$ corresponding to this particle using the relations $p^\star_{t-1} = a^{p^\star_t}_t$ and $u^\star_{t-1} = \tilde{u}^{p^\star_{t-1}}_{t-1}$.

(c) Replace the old trajectory $u_{t_0:t_1}$ by the new trajectory $u^\star_{t_0:t_1}$.

The trajectory $\mathbf{s}$ between positions t_0 and $t_1 > t_0$ ($s_{t_0:t_1}$) is updated using a similar algorithm in which k_s and q_s replace k_u and q_u.

If $u_{t_1+1} > 0$ all the new trajectories created by this algorithm will have a breakpoint between position t_1 and $t_1 + 1$. This imposes some constraints on the choice of the block $u_{t_0:t_1}$ on which the algorithm is applied in order to fulfill the reversibility condition required for an MCMC algorithm. In practice, we select blocks that have breakpoints on their last position (between t_1 and $t_1 + 1$) or that ends in a non-expressed region ($u_{t_1+1} = 0$). These blocks are obtained by picking randomly (uniformly, except for the first position to which a greater probability is attributed) the position t_0 and setting $t'_1 = t_0 + l$ where l is the integer part of a gamma distributed random variable with shape $\min\{5/(1-\alpha), P\}$ (or $\min\{5/(1-\alpha_s), P\}$ when updating $\mathbf{s}$) and scale 1 (this choice prevents the length of the block to be much longer than P as the updates become then less efficient due to degeneracy of the sampled trajectories). Then, for each segment (t_0, t'_1) we update the block (t_0, t_1) where t_1 is the last breakpoint (a shift or a drift) or the last position where $u_t = 0$ before t'_1. With this procedure a given block has the same probability to be selected for update after and before the update which warrants global reversibility provided that each update of a given block is itself reversible (e.g. verifies detailed balance condition).

In our MCMC sweeps, the number of particles in the CSMC was set to $P = 150$, excepted the validation runs presented in subsection 3.4 which relied only on 70 particles, and the number of randomly selected blocks was set to L/P where L is the length of the sequence.

2.2 Proposals for u and s SMC updates

The efficiency of an SMC algorithm, here the mixing of the Particle Gibbs, depends heavily on the choice of an appropriate proposal kernel. In this work we relied on two position-specific proposals $q_{u,t}$ and $q_{s,t}$ that aim at drawing new values of u and s near their target posterior distributions. In practice we make the proposal dependent of the observed read counts.

Our proposal $q_{u,t}$ is

$$\begin{aligned} q_t(u_{t+1}; u_t) &= \mathbf{1}_{\{u_t=0\}} \cdot \big[\frac{9}{10} \cdot \delta_0(u_{t+1}) + \frac{1}{10}\mathcal{E}(u_{t+1}; rate = \zeta_p) \\ &\quad + \mathbf{1}_{\{u_t>0\}} \cdot \big[\frac{9}{10} \cdot \delta_{u_t}(u_{t+1}) + \frac{1}{10} \cdot l_t(u_{t+1})\big] , \end{aligned}$$

where the density $l_t(u_{t+1})$ is itself a mixture that writes

$$\begin{aligned} l_t(u_{t+1}) &= \frac{1}{3} \cdot \delta_0(u) + \frac{1}{3} \cdot \Gamma(u_{t+1}, shape = \sum_{i=1:k} y_{t+i} + 1, scale = \frac{1}{k + \zeta_p}) \\ &\quad + \frac{1}{3} \cdot \mathcal{E}(u_{t+1}; rate = \zeta_p) . \end{aligned}$$

The shape and scale parameters of the Gamma distribution were derived from exact computation for an approximate (instrumental) model that assumes a expression level u_{t+1} draw from

an exponential with rate ζ_p, $u_{t+i} = u_{t+1}$ and $\nu_{t+i} = 1$ for $i = 1 : k$. In practice we set $\zeta_p = 0.1$ and $k = 3$.

Our proposal $q_{s,t}$ is

$$\begin{aligned} q_{s,t}(s_{t+1}; s_t) &= \frac{9}{10} \cdot \delta_{s_t^i}(s_{t+1}^i) + \frac{1}{20} \cdot \Gamma(s_{t+1}; shape = \sum_{i=1:k} y_{t+i} + 1, scale = \frac{1}{k+1}) \\ &\quad + \frac{1}{20} \cdot \mathcal{E}(s_{t+1}; rate = 1)\,. \end{aligned}$$

2.3 Particle Gibbs with simultaneous update of u and s

The subsection 2.1 describes our block update Particle Gibbs for $\mathbf{u}$ and $\mathbf{s}$ according to their respective conditional distributions $\mathbf{u} \mid \mathbf{s}, \Theta, \mathbf{y}$ and $\mathbf{s} \mid \mathbf{u}, \Theta, \mathbf{y}$. It is thus possible to combine these algorithms in a more global MCMC algorithm to sample $\mathbf{u}, \mathbf{s} \mid \Theta, \mathbf{y}$ or $\mathbf{u}, \mathbf{s}, \Theta \mid \mathbf{y}$. However, it can be noticed that the distribution of the observed read count y_t depends on the product $u_t \cdot s_t$ rather than u_t and s_t taken individually. We expect thus some negative correlation between u_t and s_t in the posterior distribution.

This motivated the development of another block update Particle gibbs algorithm that can update $\mathbf{u}$ while preserving as much as possible the product $u_t \cdot s_t$ by applying simultaneous modifications to $\mathbf{s}$. For the sake of clarity let z_t denote the product $u_t \cdot s_t$. If $u_t > 0$ the model can be rewritten in terms of $\mathbf{u}$ and $\mathbf{z}$ instead of $\mathbf{u}$ and $\mathbf{s}$; the trajectory of $\mathbf{z}$ being fully determined by the trajectory of $\mathbf{u}$ and the values of z_t at breakpoint positions $\{t : bk_{s,t} = \text{'change'}\}$. The Conditional SMC update described below targets $\pi(u_{t_0:t_1} \mid u_{t<t_0}, u_{t>t_1}, z_{t_0}, z_{\{t_0<t\leq t_1 : bk_{s,t}=\text{'change'}\}}, \Theta, u_{t_0:t_1} > 0)$. The new values of $\mathbf{s}$ are obtained after updating $\mathbf{u}$ when reverting to the original parametrization of the model in terms of $\mathbf{u}$ and $\mathbf{s}$, by using the relation $s_t = z_t/u_t$ at breakpoint positions $\{t : bk_{s,t} = \text{'change'}\}$.

The procedure to select a block $u_{t_0:t_1}$ consists of: selecting randomly t'_0, searching for $t_0 = min\{t \geq t'_0 : u_{t_0} > 0\}$, setting $t'_1 = t_0 + l$ where l is a random variable with mean $\min\{5/(1-\alpha), P\}$ (see subsection 2.1), and then searching for $t_1 = \max\{t_1 \leq t'_1 : u_{t0:t1} > 0\}$. Let $t_{0,s}$ and $t_{1,s}$ denote $\max\{t \leq t_0 : bk_{s,t} = \text{'change'}\}$ and $\min\{t \geq t_1 : bk_{s,t+1} = \text{'change'}\}$, respectively.

This Conditional SMC update writes

1. For $t = t_0$,
   (a) For the first $P-1$ particles $1 \leq p < P$, draw $\tilde{u}_t^p$ from the proposal density $q_{u,t_0}(\tilde{u}_{t_0}^p; u_{t_0-1})$ where q_{u,t_0} is a position-specific proposal that uses the observed read-counts to focus sampling near the target (see subsection 2.2).
   (b) Set $\tilde{u}_t^P = u_t$
   (c) For $1 \leq p \leq P$, set $\tilde{s}_{t_0}^p = s_{t_0} \cdot u_{t_0}^p / \tilde{u}_{t_0}^p$
   (d) For $1 \leq p \leq P$, calculate the normalized particle weights

$$w_{t_0}^p \propto \frac{k_u(\tilde{u}_{t_0}^p; u_{t_0-1}) \cdot e(y_{t_0}; \tilde{u}_{t_0}^p, \tilde{s}_{t_0}^p) \cdot \mathbf{1}_{\{\tilde{u}_{t_0}^p > 0\}} \cdot \pi_s(\tilde{s}_{t_0}^p)/\tilde{u}_{t_0}^p}{q_{u,t_0}(\tilde{u}_{t_0}^p; u_{t_0-1})} \cdot \prod_{t=t_{0,s}}^{t_0-1} e(y_t; u_t, \tilde{s}_{t_0}^p)$$

   such as $\sum_{p=1:P} w_{t_0}^p = 1$. In the numerator, the term $\pi_s(\tilde{s}_{t_0}^p)/\tilde{u}_{t_0}^p$ corresponds to $\pi_z(z_{t_0} \mid \tilde{u}_{t_0})$ where $z_{t_0} = \tilde{u}_{t_0} \cdot \tilde{s}_{t_0}$. The extra term $\prod_{t=t_{0,s}}^{t_0-1} e(y_t; u_t, \tilde{s}_{t_0}^p)$ accounts for the modified distribution of the observed read count between $t_{0,s}$ and $t_0 - 1$.

2. From $t = t_0 + 1$ to $t = t_1$,

   (a) If $1/\sum_{p=1}^{P}(w_{t-1}^p)^2 > P/4$ (the effective sampling size is above $P/4$) then $a_t^p = p$ for $1 \leq p \leq P$ and $w_t'^p = w_{t-1}^p$.

   (b) If $1/\sum_{p=1}^{P}(w_{t-1}^p)^2 \leq P/4$ then the index of the ancestor particle a_t^p is drawn from a multinomial distribution with weights $(w_{t-1}^p)_{1:P}$ for $1 \leq p < P$, $a_t^P = P$ and $w_t'^p = 1$ for $1 \leq p \leq P$.

   (c) For $1 \leq p < P$, draw $\tilde{u}_t^p$ from the proposal density $q_{u,t}(\tilde{u}_t^p; \tilde{u}_{t-1}^{a_t^p})$.

   (d) Set $\tilde{u}_t^P = u_t$.

   (e) If $bk_{s,t}$ = 'no change' then for $1 \leq p \leq P$, set $\tilde{s}_t^p = \tilde{s}_{t-1}^{a_t^p}$ and calculate the normalized particle weights $w_t^p \propto w_t'^p \cdot \frac{k_u(\tilde{u}_t^p; \tilde{u}_{t-1}^{a_t^p}) \cdot e(y_t; \tilde{u}_t^p, \tilde{s}_t^p)}{q_{u,t}(\tilde{u}_t^p; u_{t-1}^{a_t^p})}$ such as $\sum_{p=1:P} w_t^p = 1$.

   (f) If $bk_{s,t}$ = 'change' then, for $1 \leq p \leq P$, set $\tilde{s}_t^p = s_t \cdot u_t^p / \tilde{u}_t^p$ and calculate the normalized particle weights $w_t^p \propto w_t'^p \cdot \frac{k_u(\tilde{u}_t^p; \tilde{u}_{t-1}^{a_t^p}) \cdot e(y_t; \tilde{u}_t^p, \tilde{s}_t^p) \cdot \mathbf{1}_{\{\tilde{u}_t^p > 0\}} \cdot \pi_s(\tilde{s}_t^p)/\tilde{u}_t^p}{q_{u,t}(\tilde{u}_t^p; \tilde{u}_{t-1}^{a_t^p})}$ such as $\sum_{p=1:P} w_t^p = 1$. The extra term in the numerator $\pi_s(\tilde{s}_t^p)/\tilde{u}_t^p$ corresponds to $\pi_z(z_t \mid \tilde{u}_t)$ where $z_t = \tilde{u}_t \cdot \tilde{s}_t$.

3. For $t = t_1$,

   (a) Sample the index $p_{t_1}^\star$ of particle corresponding to the new trajectory at position t_1. Each particle p has a probability proportional to $w_{t_1}^p \cdot k_u(u_{t_1+1}; \tilde{u}_{t_1}^p) \cdot \prod_{t=t_1+1}^{t_{1,s}} e(y_t; u_t, \tilde{s}_{t_1}^p)$.

   (b) Backtrace the whole trajectory from $t = t_1$ to $t = t_0$ corresponding to this particle using the relations $p_{t-1}^\star = a_t^{p_t^\star}$, $u_{t-1}^\star = \tilde{u}_{t-1}^{p_{t-1}^\star}$, $s_{t-1}^\star = \tilde{s}_{t-1}^{p_{t-1}^\star}$.

   (c) Replace the old trajectories $u_{t_0:t_1}$ and $s_{t_0:t_1}$ and by the new trajectories $u_{t_1:t_0}^\star$ and $s_{t_1:t_0}^\star$. Propagate the modifications in $\mathbf{s}$ down to position $t_{0,s}$ and up to position $t_{1,s}$ by setting $s_t = s_{t_0}^\star$ for $t_{0,s} \leq t < t_0$ and $s_t = s_{t_1}^\star$ for $t_1 < t \leq t_{1,s}$.

3 Estimation of the parameters

3.1 Estimation of emission and local correlation parameters

The Bayesian framework adopted in this work allows in principle the estimation of the complete set of parameters within the same MCMC algorithm. However, most of the parameters of the emission model (*i.e.* κ, θ, κ_s and α_s) can be estimated directly from the characteristics of the read-counts without hidden path reconstruction. We choose here to estimate these parameters beforehand in order to improve the MCMC mixing with respect to the other parameters.

The estimation of the parameters κ and θ relies on positive read counts occurring in regions with with very low coverage. Here we considered counts in a context with less than 1 position with positive read count in 100 bp. At such a position t, the observed count is very likely issued from a single initial molecule ($x_t = 1$) because u_t is very small. We can thus estimate κ and θ directly from the distribution of these read counts in low expression regions as they arise from the conditional distribution $(y_t \mid x_t = 1, y_t > 0)$ whose density writes

$$\pi(y_t \mid x_t = 1, y_t > 0) = \frac{\mathcal{NB}(y_t; \kappa, \frac{\theta}{\theta+1})}{1 - \left(\frac{1}{\theta+1}\right)^{\kappa}} .$$

In practice we compute the estimates of the first (μ_1) and second (μ_2) moments of $(y_t \mid x_t = 1, y_t > 0)$ and we derive moment estimates of κ and θ by solving the equations

$$\frac{\frac{\kappa}{\kappa+1} \cdot (\frac{\mu_2}{\mu_1} - 1)}{1 - \left(\frac{\kappa+1}{\frac{\mu_2}{\mu_1}+\kappa}\right)^{\kappa}} - \mu_1 = 0$$

and

$$\theta = \frac{\frac{\mu_2}{\mu_1} - 1}{\kappa + 1} .$$

We show in fig. S2 (leftmost plots) the fit between the empirical distribution of $(y_t \mid x_t = 1, y_t > 0)$ and our NB model. At higher coverage, the analysis of the relationships variance versus mean and fraction of zero-counts fraction versus mean shows that both characteristics are not well captured if $s_t = 1$ ($V(s_t) = 0$ and $\kappa_s = +\infty$) (fig S2, middle plots).

After integration over the possible values of s_t and neglecting background noise and outliers, the density of the marginal distribution $(y_t \mid u_t)$ writes

$$\pi(y_t \mid u_t) = \sum_{x_t \geq 0} \mathcal{NB}(x_t; \kappa_s, \frac{u_t}{\kappa_s \kappa \theta + u_t}) \cdot \mathcal{NB}(y_t; \kappa, \frac{x_t \theta}{x_t \theta + 1}) .$$

We can thus also derive the variance and the fraction of zero counts given u_t and the parameters of the model

$$\begin{aligned} \mathbb{V}(y_t \mid u_t) &= (\frac{1}{\kappa} + \frac{1}{\kappa_s} + \frac{1}{\kappa_s \kappa}) \cdot u_t^2 + u_t \cdot (1 + \theta + \kappa\theta) \\ \mathbb{P}(y_t = 0 \mid u_t) &= \sum_{x_t \geq 0} \mathcal{NB}(x_t; \kappa_s, \frac{u_t}{\kappa_s \kappa \theta + u_t}) \cdot \mathcal{NB}(0; \kappa, \frac{x_t \theta}{x_t \theta + 1}) \end{aligned}$$

In order to use these two relationships to estimate κ_s we rely on chromosome segments inside which we expect that the expression level u_t should be homogeneous (fig. S2, middle plots). We delineate these segments as the central part of the open reading frames (ORFs, regions

without stop codons in a particular reading frame) that are too long to occur by chance and that correspond thus very likely to coding sequences. In practice, we select ORFs above 400 bp and we discard the first and last 100 bp to account for uncertainty on the position of the start codon and coverage effects that can occur at the borders of the transcripts. For each of these ORF segments, indexed by g, we compute the mean of the read counts (u_g), the variance of the reads counts (σ_g^2) and the fraction of zero-counts (f_g^0). Two estimates of κ_s are then obtained

$$\begin{aligned}\hat{\kappa}_s^{\sigma} &= \underset{\kappa_s}{argmin} \sum_g \frac{|\mathbb{V}_{\kappa_s}(y_t \mid u_t = u_g) - \sigma_g^2|}{\sigma_g^2} \\ \hat{\kappa}_s^{0} &= \underset{\kappa_s}{argmin} \sum_g |\mathbb{P}_{\kappa_s}(y_g = 0 \mid u_t = u_g) - f_g^0| \,,\end{aligned}$$

and the final estimate is taken as the average of both estimates

$$\kappa_s = \frac{\hat{\kappa}_s^{\sigma} + \hat{\kappa}_s^{0}}{2} \,.$$

.

The parameter α_s is adjusted to account for the autocorrelation of read counts observed inside the ORF segments (fig. S2, rightmost plots). Under the assumption of our model for s_t, and assuming constant expression level u_g the autocorrelation writes

$$cor_g(y_t, y_{t+l}) = (\frac{u_g^2 \cdot \frac{1}{\kappa_s}}{\sigma_g^2}) \cdot \alpha_s^l \,,$$

where g is the index of the segment with homogeneous expression u_g. This allows to estimate α_s with a linear regression across segments (g) and lag values (l). In practice, we carry this estimation on the ORFs with length above 2000 bp and expression level above 0.5 reads/bp and on the lag values $1 \leq l \leq 10$.

3.2 Bayesian estimation of parameters

Estimation of parameters characterizing the dynamics of the expression level path parameters was carried out within a Bayesian framework. The following priors were used for the different parameters:

- $(\alpha, \gamma_u, \gamma_d, \beta, \beta_0) \sim Dirichlet(concentration\ parameters = 100, 1, 1, 1, 1, 1)$,
- $\eta \sim Beta(shape_1 = 1, shape_2 = 100)$,
- $\zeta \sim Exponential(rate = 1)$,
- $(1 - \epsilon_b - \epsilon_o, \epsilon_b, \epsilon_o) \sim Dirichlet(concentration\ parameters = 100, 1, 1)$

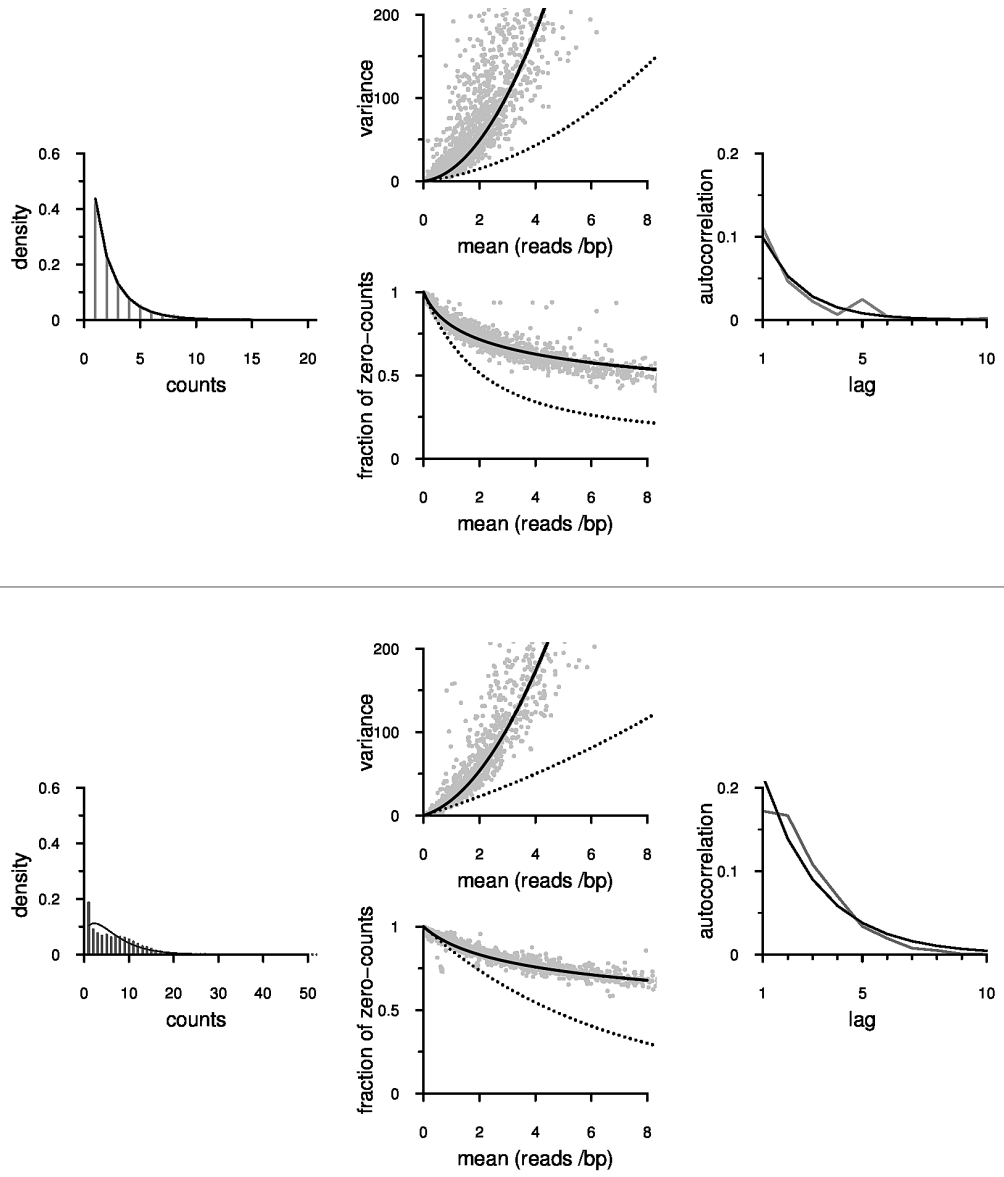

Figure S2 Estimation of the parameters κ, θ, κ_s and α_s on the *S. cerevisie* data set SRR121907 (top panel) and *E.coli* data set SRR794838 (bottom panel). The first step consists of estimating of "amplification" parameters κ and θ ($\mu_a = \kappa\theta$) from the distribution of isolated positive counts, i.e. positions with positive counts issued from low expression regions (leftmost plots). Our model for these counts corresponds to a negative binomial with size κ and probability $\frac{\theta}{\theta+1}$ whose fit is shown (leftmost plots, black line). The second step consists of estimating the parameter κ_s that allows to account for the overdispersion that remains unexplained with κ and θ as well as the fraction of zero counts (middle plots). For each ORF we plot variance versus mean and fraction of zero counts versus mean (middle plots - grey points). We superimpose onto these plots the relationships captured with the estimated value of κ_s (middle plots, black lines) and the relationships captured before taking into account s_t, *i.e.* assuming $s_t = 1$ ($\kappa_s = +\infty$, $\mathbb{V}(s_t) = 0$) (middle plots - dotted lines). The third step consists of estimating α_s to account for the autocorrelation of read counts observed inside the ORFs (rightmost plots). We show here the average of autocorrelations (lags from 1 to 10) on all the ORFs (leftmost plot, gray line) and the autocorrelation captured with the estimated value of α_s (leftmost plots, black line).

3.3 Parseq MCMC algorithm

Each sweep of our MCMC algorithm to sample the joint distribution

$$\mathbf{u}, \mathbf{s}, \mathbf{bk_u}, \mathbf{bk_s}, \mathbf{o}, \underbrace{\alpha, \gamma_u, \gamma_d, \beta, \beta_0, \eta, \zeta, \epsilon_o, \epsilon_b,}_{\Theta} \mid \mathbf{y}$$

consists of updating one or a subset of the variables ($\mathbf{u}$, $\mathbf{s}$ and the parameters) according to their conditional distribution (Gibbs sampler). Namely, one sweep of the algorithm is composed of:

- Update $\mathbf{u}$ preserving $\mathbf{u} \mid \mathbf{s}, \Theta, \mathbf{y}$ with the CSMC algorithm described in subsection 2.1.
- Update $\mathbf{s}$ preserving $\mathbf{s} \mid \mathbf{u}, \Theta, \mathbf{y}$ with the CSMC algorithm described in subsection 2.1.
- Simultaneous update $\mathbf{u}$ and $\mathbf{s}$ preserving $\mathbf{u}, \mathbf{s} \mid \Theta, \mathbf{y}$ with the CSMC algorithm described in subsection 2.3
- Update of $\mathbf{o}$ according to $\mathbf{o} \mid \mathbf{u}, \mathbf{s}, \Theta, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of $\mathbf{bk_u}$ according to $\mathbf{bk_u} \mid \mathbf{s}, \mathbf{u}, \mathbf{o}, \Theta, \mathbf{y}$ (the variables written in green do not play a role in the conditional distribution). This conditional distribution is sampled directly (Gibbs-type update).
- Update of $\mathbf{bk_s}$ according to $\mathbf{bk_s} \mid \mathbf{s}, \mathbf{u}, \mathbf{o}, \mathbf{bk_u}, \Theta, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update, that consists simply of differentiating 'move' and 'no-move' based on $\mathbf{s}$).
- Update of $(\alpha, \gamma_u, \gamma_d, \beta, \beta_0)$ according to $\alpha, \gamma_u, \gamma_d, \beta, \beta_0 \mid \mathbf{s}, \mathbf{u}, \mathbf{o}, \mathbf{bk_u}, \mathbf{bk_s}, \eta, \zeta, \epsilon_o, \epsilon_b, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of η according to $\eta \mid \mathbf{s}, \mathbf{u}, \mathbf{o}, \mathbf{bk_u}, \mathbf{bk_s}, \alpha, \gamma_u, \gamma_d, \beta, \beta_0, \zeta, \epsilon_o, \epsilon_b, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of ζ according to $\zeta \mid \mathbf{s}, \mathbf{u}, \mathbf{o}, \mathbf{bk_u}, \mathbf{bk_s}, \alpha, \gamma_u, \gamma_d, \beta, \beta_0, \eta, \epsilon_o, \epsilon_b, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of (ϵ_o, ϵ_b) according to $\epsilon_o, \epsilon_b \mid \mathbf{s}, \mathbf{u}, \mathbf{o}, \mathbf{bk_u}, \mathbf{bk_s}, \alpha, \gamma_u, \gamma_d, \beta, \beta_0, \eta, \zeta, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).

3.4 Algorithm validation

The implementation of complex MCMC algorithms such as the one described above (subsection 3.3) is error prone. For this reason we implemented a validation strategy to check the exactness of MCMC algorithm. This strategy relies on the idea of running an extended version of MCMC in which a step where the observations $\mathbf{y}$ aresampled from their conditional distribution $\mathbf{y} \mid \mathbf{u}, \mathbf{s}, \boldsymbol{\Theta}$ is added to each sweep. The algorithm samples then the joint distribution $\mathbf{y}, \mathbf{u}, \mathbf{s}, \boldsymbol{\Theta}$ instead of $\mathbf{u}, \mathbf{s}, \boldsymbol{\Theta} \mid \mathbf{y}$. If the algorithm is exact the sampled values of $\boldsymbol{\Theta}$ should thus corresponds exactly to the prior distribution (which is the marginal of the joint distribution $\mathbf{y}, \mathbf{u}, \mathbf{s}, \boldsymbol{\Theta}$).

For faster convergence the extended algorithm was run on a short sequence (500 bp), and the output values for $\boldsymbol{\Theta}$ were compared to the prior. The results are shown Figure S3.

3.5 Impact of SMC exactness on MCMC output

Our final algorithm relies on Conditional SMC updates of the hidden trajectories $\mathbf{u}$ and $\mathbf{s}$ which provide an exact MCMC algorithm (termed Particle Gibbs) for sampling their distribution given the observed $\mathbf{y}$. However in preliminary versions of this work we relied on a simpler algorithm that consisted in forward SMC filtering and subsequent backtracking of the trajectory of one sample particle. This simpler algorithm provides a trajectory that is only approximately distributed according to the target conditional distribution (higher number of particles in the SMC filtering increase the precision). To illustrate the impact of using an approximate algorithm (MCMC based on SMC filtering) instead of an exact algorithm (Particle Gibbs based on Conditional SMC) we compared the parameter values sampled using the extended versions of both algorithms. The results shown in Figure S4 indicates a markedly biased behavior of the approximate MCMC algorithm.

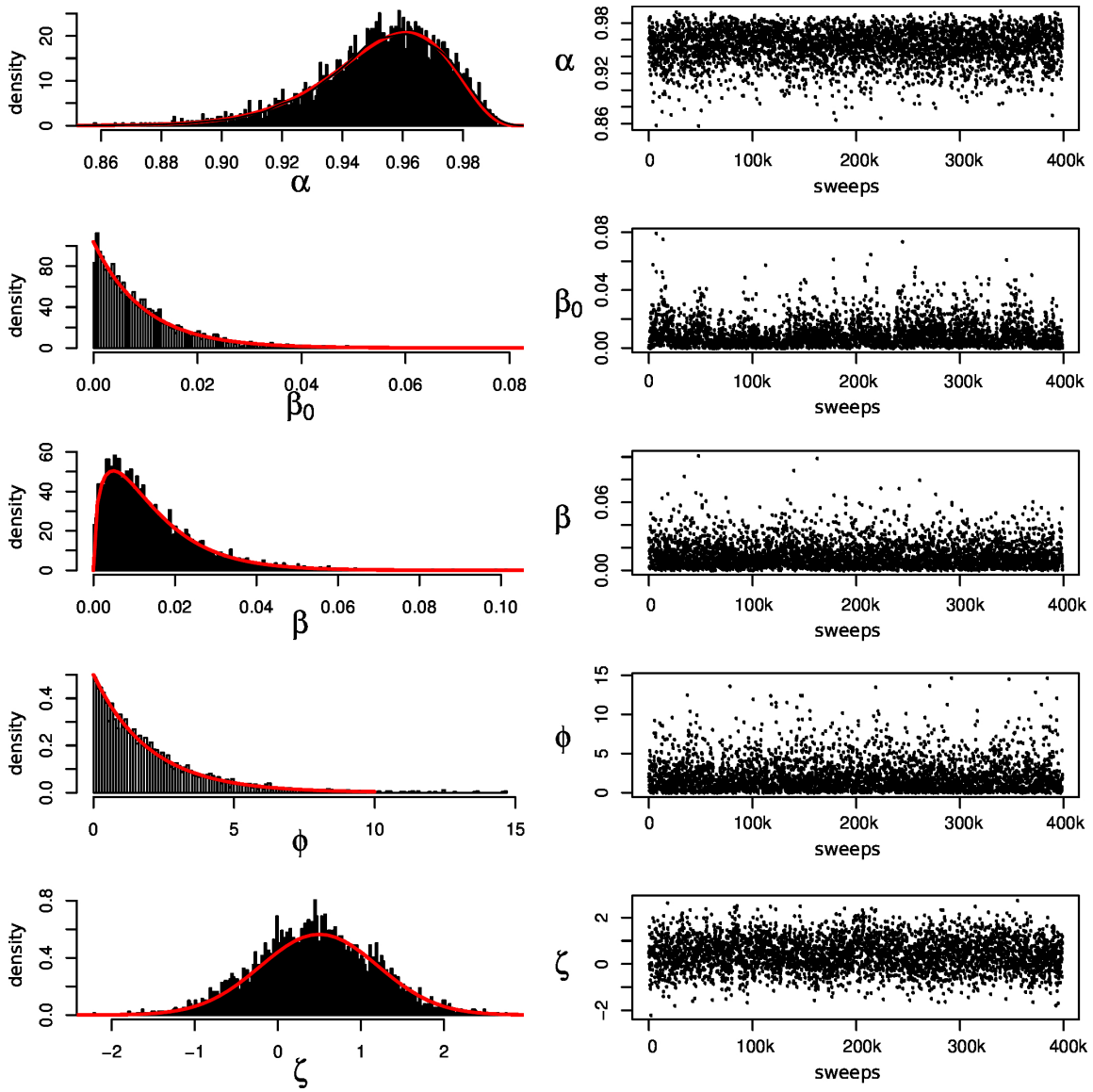

Figure S3 Validation of the MCMC algorithm. For the emission model we used a Negative Binomial (overdispersion ϕ). We show the histogram (left column) and the convergence plot (right column) for the parameter values sampled with the extended version of the MCMC including a step where the observations $\mathbf{y}$ are sample sampled from their conditional distribution $\mathbf{y} \mid \mathbf{u}, \mathbf{s}, \boldsymbol{\Theta}$ is added to each sweep. If the implementation of the algorithm does not contain errors the sampled values should be conform to the parameter priors which is indeed the case (densities of the parameter priors are shown in red). The thinning step applied here is 100 sweeps, the sequence length is 500 bp and we use 70 particles.

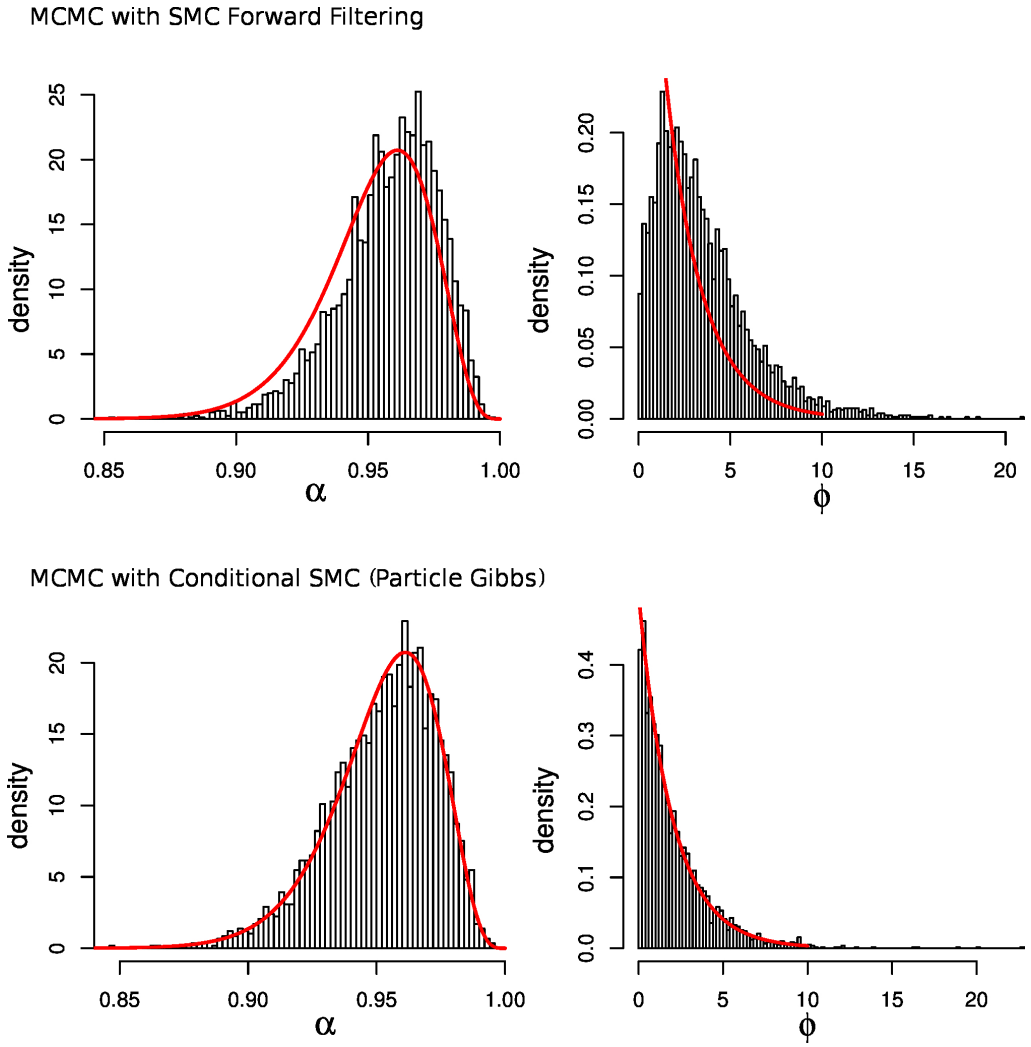

Figure S4 Comparison of MCMC outputs based on SMC filtering and on Conditional SMC (approximate vs. exact MCMC). For both approaches we used a Negative Binomial (overdispersion ϕ) as emission distribution. The extended MCMC algorithm that allows, in principle, to sample the prior distribution of the parameters (red line) is run on a sequence of length 500 bp. The SMC filtering update relies here on 500 particles whereas the Conditional SMC algorithm uses only 70 particles. The approximate MCMC algorithm based on SMC filtering exhibits a markedly biased behavior (top) whereas the MCMC algorithm based on Conditional SMC (Particle Gibbs) provides a sample that conform to the prior (bottom). Here ϕ corresponds to the overdispersion of the negative binomial used as emission model in our preliminary analyses.

4 Results on *S. cerevisiae* and *E. coli* data-sets

4.1 Simulation of synthetic data

Our synthetic datasets were generated based on the sequence and annotation of both strands of *Saccharomyces cerevisiae* chromosome IV with Flux Simulator vs 1.2 (Griebel *et al.*, 2012). RNA-seq reads of length 50 bp were simulated specifying uniform RNA fragmentation. We allowed no variability in TSS and pA positions to be able to accurately assess the performance of transcript border detection (parameters TSS_MEAN and POLYA_SCALE set to NaN). Due to the lack of variability in transcript borders and to Flux Simulator fragmentation model, strong read count peaks at transcript borders were obtained with default fragmentation parameters. To modify this unrealistic behavior we increased the fragmentation rate up to an average fragment length of 20 (FRAG_UR_D0 10, FRAG_UR_DELTA 1, FRAG_UR_ETA 20) before size selection. After size-selection lengths are normally distributed around length 100 (standard deviation 2).

We analyzed Parseq and Cufflinks performance on increasing sequencing depth. We increased the depth by: i) keeping constant the initial number of mRNA molecules and varying amplification parameters (scenario 1) and ii) keeping constant the amplification coefficient and varying the initial number of mRNA molecules (Table S1). For the scenario 1 we varied the reads to molecules ratio from 5:1 to 80:1 by keeping fixed the number of molecules to 30.000. For scenario 2 we kept a constant reads to molecules ratio of 20:1 and varied the number of molecules from 7.500 to 120.000 . The sequencing depth increased similarly in both scenarios from 150k to 2400k reads.

Scenario 1		Scenario 2		
# mRNA mol.	# of reads per mol.	# mRNA mol.	# of reads per mol.	# of reads
30000	5:1	7500	20:1	150000
30000	10:1	15000	20:1	300000
30000	20:1	30000	20:1	600000
30000	40:1	60000	20:1	1200000
30000	80:1	120000	20:1	2400000

Table S1 Two scenarios for increasing sequencing depth in synthetic data. Flux Simulator values for the initial number of mRNA molecules and final read counts in scenarios 1 and 2.

4.2 Parameter estimates

The parameters were estimated with Parseq MCMC algorithm. Preliminary runs indicated that it is difficult to estimate simultaneously the frequency (γ_u and γ_d) and the amplitude (λ_u and λ_d) of the drifts (slow convergence behavior typical of a flat likelihood function). This is not really surprising as several small amplitude drift moves can be difficult to distinguish from one drift move of larger amplitude. Therefore, we fixed $\lambda_u = \lambda_d = 5.0$ which corresponds to a drift average relative amplitude 0.2.

The Table S2 summarizes the results obtained on *S. cerevisie* SRR121907 and E. *coli* SRR794838 datasets. The Figure S5 illustrates the convergence (parmeters) of the algorithm.

	S. cerevisiae		*E. coli*	
parameter	mean$^{(a)}$	sd.$^{(b)}$	mean$^{(a)}$	sd.$^{(b)}$
parameters of the read-count emission model				
κ	0.61	-	1.91	-
θ	1.93	-	3.22	-
ϵ	0.00067	0.0014	0.0019	0.00042
ϵ_o	0.0000020	0.0000043	0.0000011	0.0000012
transition kernel for the local scaling variable **s**				
α_s	0.53	-	0.64	-
κ_s	0.31	-	0.20	-
transition kernel for the expression level **u**				
α	0.97	0.0071	0.97	0.0058
γ_u	0.011	0.0033	0.014	0.011
γ_d	0.013	0.0038	0.018	0.012
β_0	0.00060	0.00011	0.00056	0.00016
β	0.00080	0.00016	0.00047	0.0000090
η	0.00072	0.00012	0.00080	0.00016
ζ	1.18	0.28	0.70	0.22
λ_u, λ_d (fixed)	5.00	-	5.00	-

$^{(a)}$ mean and $^{(b)}$ standard-deviation across the 1 Mbp subdivisions of the genome.

Table S2 Parseq parameters estimated on *S. cerevisie* SRR121907 and E. *coli* SRR794838 data-sets. Parameter estimates are first obtained separately for each chromosome subdivision of $\approx 1Mb$ with the Parseq MCMC algorithm. We averaged the sampled values after discarding $1/10^{th}$ of the sweeps (burn-in). Parameter estimates are then averaged between genome fragments to obtain the final set of parameters used for expression level reconstruction (column 'mean').

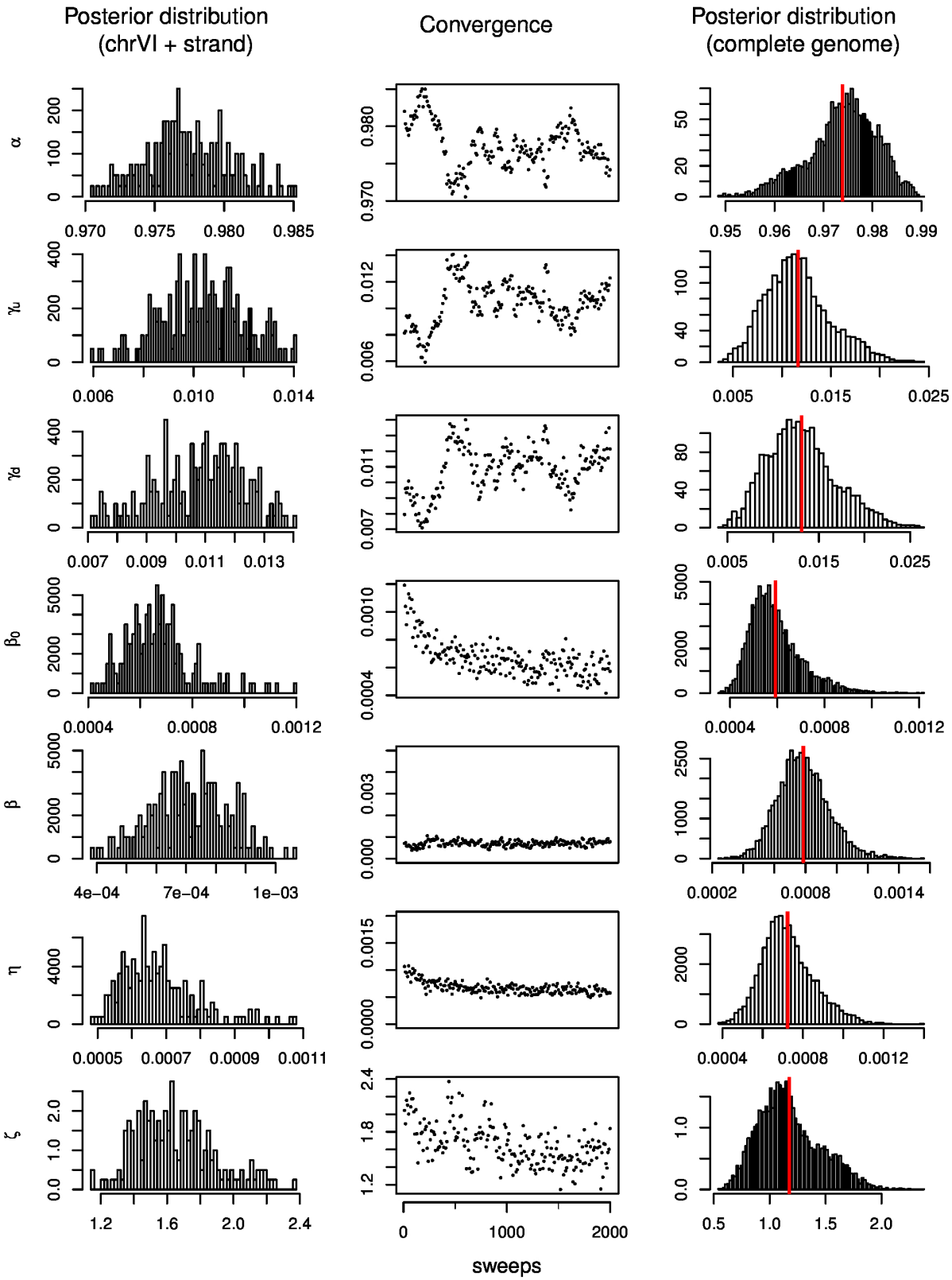

Figure S5 Parameter estimation on SRR121907 dataset(*S. cerevisiae*): chromosome VI strand+ (left histogram and convergence plot) and complete genome (right histogram). Left and right columns: histograms of the sampled values approximating the marginal posterior distributions. The complete genome histogram includes sampled values from for all chromosomes and both strands. Middle column: convergence plot along 2000 sweeps of the MCMC algorithm, excluding a 200 sweeps burn-in; parameter values are plotted every 10 sweeps.

4.3 Settings of the MCMC runs

As explained in the main text (see also subsection 4.2) the Parseq work-flow includes four steps (steps 1-3 and postprocessing). The second and third steps involve MCMC runs.

In the second step, Parseq MCMC algorithm is run on the whole data set (but independently on each chromosome fragment of $\approx$ 1Mb) in order to obtain estimates of the transcription dynamics parameters. Each CSMC update relied on 150 particles. For both real and simulated data sets the MCMC algorithm was run for 2200 sweeps (1 sweep on a sequence of 1 Mbp takes 0.8-1.2 min on a CPU Intel Core i7-3610QM CPU @ 2.30GHz); the thinning step was set to 10 and the first 200 sampled values discarded (burn-in).

In the third step, Parseq MCMC algorithm is run on the whole data set (on each genome fragment) with fixed parameters in order to obtain a sample from the posterior distribution of $\mathbf{u}$ given $\mathbf{y}$ and Θ. As in the second step, CSMC update relied on 150 particles and the MCMC was consisted of $2,200$ sweeps in order to generated 200 trajectories after thinning (1/10) and burn-in (200 sweeps).

4.4 Choice of a cut-off on expression level

We noticed in our estimates of **u** that many regions outside annotated genes are associated with a low but non-null expression level. Genuine pervasive transcription and background noise of the technology could contribute to this low expression signal. There are many situations where, whatever the origin (biology or artifact) of this low expression signal, we can be interested in predicting a set of expressed segments that does not contain these regions with low but non-null expression signal. To select an appropriate cut-off for expression level thresholding we examined the marginal distribution of the estimated expression levels **u** (Figure S6). For both data sets we observe an accumulation of position associated with low expression inconsistent with the exponential distribution. In practice, we set our expression cut-off to 0.1 reads/bp for *S. cerevisiae* and 0.25 reads/bp for *E. coli* when calling transcripts and breakpoints.

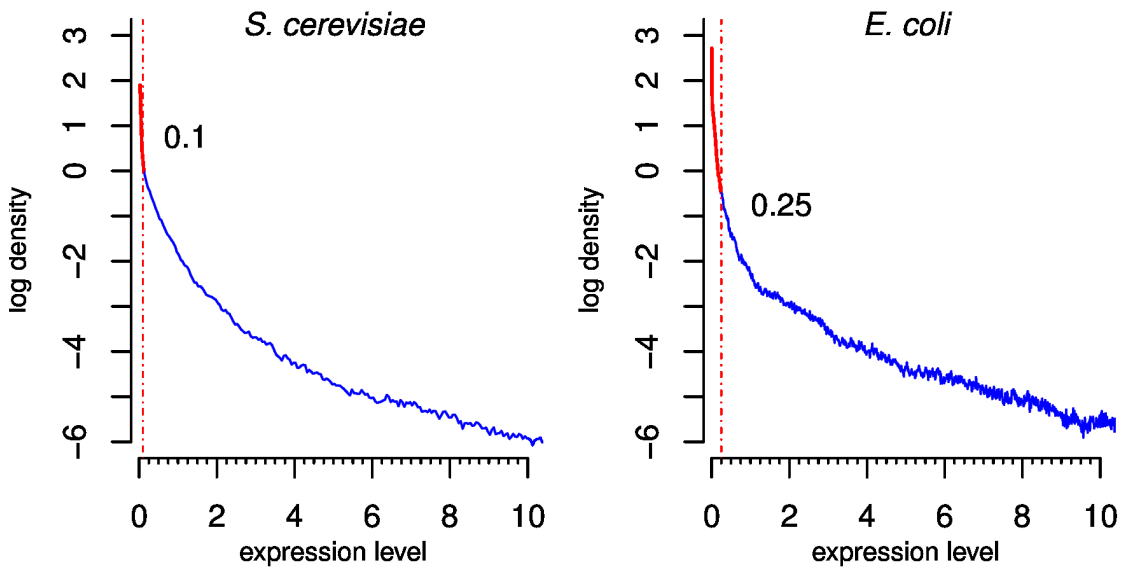

Figure S6 Marginal distribution of the estimated expression levels **u** on *S. cerevisiae* (SRR121907) and *E. coli* (SRR794838) data sets. In this log-density *vs.* expression level plot an exponential distribution corresponds to a straight-line, as observed in the left part of the blue segments. Below 0.1 reads/bp in *S. cerevisiae* and 0.25 reads/bp in *E. coli* a sharp accumulation is visible (red segment). The cut-off on expression level that served for calling transcripts and breakpoints were set to these values (vertical dotted red line).

4.5 Local score approach for breakpoint posterior post-processing

Due to residual uncertainty on the exact breakpoint positions, the posterior position-specific breakpoint probability could not be used directly to establish breakpoint predictions. It is indeed necessary to cluster the adjacent positions that could correspond to a same breakpoint. We used this purpose a local score approach.

Our local score was defined by the classical recurrence relation $s_t = \max\{s_{t-1} + z_t - m, 0\}$, where s_t is the score at position t, z_t is the signal in which we search enriched regions, m is a penalty greater than the average of z_t. In our context, $z_t = \pi(bk_{u,t} = \text{'shift'}, u_t > c \mid \mathbf{y}, \Theta)$ where c is the cut-off on expression level (see subsection 4.4). We selected the local maximal

scoring segment in each region of positive score and the score was reset to 0 after the end position of each of these segments to avoid overlooking downstream high scoring segments that would not be separated by gaps of positions with score 0. In practice m was set to 0.005 and c to 0.5.

For each high scoring segment defined by the above procedure, we computed the cumulated probability of breakpoint $\sum_{t=t_1}^{t_2} \pi(bk_{u,t} = \text{'shift'}, u_t > c \mid \mathbf{y}, \Theta)$ between the segments end-points t_1 and t_2. We used this cumulated probability as a confidence value of the breakpoint prediction. Segments with cumulated probability greater than 1 and therefore corresponding to more than one breakpoint were divided in subsegments with equal and subunitary cumulated probability.

Finally, for each segment, a point-estimate of the position of the breakpoint was obtained as the segment mid-point in terms of cumulated probability.

4.6 Supplementary results on prediction accuracy

We report in this subsection some results that complement those presented in Table 1 of the main text. In Figure S7 we show how the choice of the expression cut-off modifies the results. It allows also a thorough comparison of the predictions made by Parseq and Cufflinks independently of the particular value taken for this cut-off. In Table S3 we show the impact of being more strict on the maximum allowed distance for matching the predictions with the reference annotations: $\pm$ 25 bp here, instead of $\pm$ 50 bp in Table 1. In particular, it appears that the choice of a particular distance cut-off does not modify the relative ranking of the three approaches: Parseq, Cufflinks and Rockhopper (see main text for details on how we evaluated Rockhopper in a de novo prediction context).

	S. cerevisiae			*E. coli*			
	Reference	Parseq	Cufflinks	Reference	Parseq	Cufflinks	Rockhopper
Trans.							
Sens.	CDSs & UTRs	0.83 (0.91)	0.83 (0.87)	Operons	0.56 (0.81)	0.6 (0.75)	0.21 (0.39)
PPV	CDSs & UTRs	0.9 (0.68)	0.9 (0.81)	Operons	0.76 (0.57)	0.72 (0.61)	0.91 (0.86)
5' ends							
Number		6689 (8353)	5484 (13622)		1846 (2193)	1577 (7962)	2949 (4401)
Sens.	TSSs	0.4 (0.4)	0.27 (0.28)	Promoters	0.17 (0.18)	0.1 (0.14)	0.06 (0.12)
PPV	TSSs & 5'UTRs	0.37 (0.3)	0.36 (0.16)	P. & O. 5'-ends	0.28 (0.25)	0.24 (0.07)	0.16 (0.16)
3' ends							
Number		6287 (7440)	5484 (13622)		1327 (1342)	1577 (7962)	2949 (4401)
Sens.	pAs	0.39 (0.4)	0.27 (0.28)	Terminators	0.07 (0.06)	0.03 (0.06)	0.02 (0.04)
PPV	pAs & 3'UTRs	0.42 (0.37)	0.37 (0.16)	T. & O. 3'-ends	0.22 (0.19)	0.14 (0.05)	0.04 (0.06)

Table S3 Detection of transcribed positions and transcript borders on *S. cerevisiae* (SRR121907) and *E. coli* (SRR794838) data-sets. Predictions and reference data were matched based on a $\pm$25 bp distance cut-off, instead of $\pm$50 bp in Table 1 of the main text. Outside parentheses: results obtained after applying a stringent expression cut-off. *S. cerevisiae*: 0.1 reads/bp for Parseq, 100 fragments per transcript for Cufflinks. *E. coli*: 0.25 reads/bp cut-off for Parseq, 200 fragments per transcript for Cufflinks, $z = 0.2$ for Rockhopper. Between parentheses: 0^+ reads/bp for Parseq, 5 fragments per transcript for Cufflinks, and $z = 0.01$ for Rockhopper.

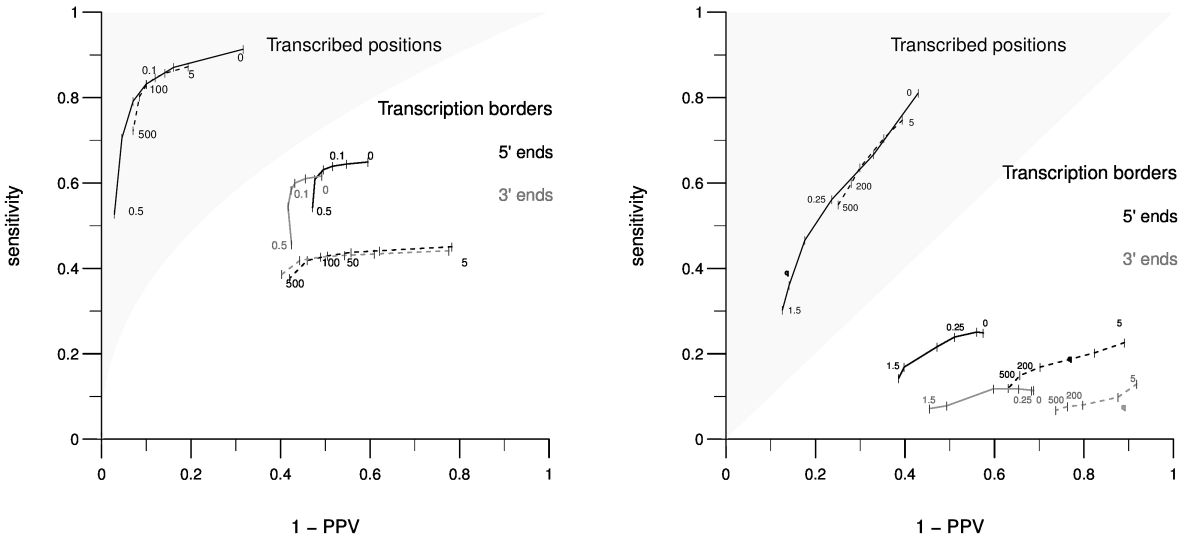

Figure S7 Impact of varying the expression cut-off on the accuracy of predictions. Results are shown for the *S. cerevisiae* data-set SRR121907 (left panel) and *E. coli* data-set SRR794838 (right panel). The performance obtained for three types of features are reported on the same plot: transcribed positions (gray area, upper right corner), transcript 5'-ends (black lines) and 3'-ends (gray lines). The results of Parseq for expression cut-offs increasing from 0^+ to 0.5 reads/bp for *S. cerevisiae* and 0^+ to 1.5 reads/bp for *E. coli* are represented by plain lines. The results of Cufflinks for minimum fragments required per transcripts increasing from 5 to 500 are represented by dashed lines. For *E. coli* we whow also results for Rockhopper with $z = 0.2$ (bullet points).

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(3), 269–342.

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: fifteen years later. *The Oxford Handbook of Nonlinear Filtering*, pages 656–704.

Griebel, T., Zacher, B., Ribeca, P., *et al.* (2012). Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Research*, **40**(20), 10073–10083.