Transcriptome analysis from high-throughput sequencing count data

Bogdan Mirauta

UPMC
PARISUNIVERSITAS

THÈSE DE DOCTORAT DE l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Bogdan MIRĂUŢĂ

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Etude du transcriptome à partir de données de comptages issues de séquençage haut débit

~

Transcriptome Analysis from High-Throughput Sequencing Count Data

soutenue le 12 décembre 2014 devant le jury composé de :

M. Christophe Ambroise	Rapporteur
M^{me}. Alessandra Carbone	Directrice de thèse
M. Nicolas Chopin	Examinateur
M. Pierre Nicolas	Encadrant
M. Hugues Richard	Encadrant
M. Claude Thermes	Examinateur
M. Jean-Daniel Zucker	Examinateur

et suite aux rapports delivrés par M. Christophe Ambroise et M. Jean-Philippe Vert

Acknowledgements

Quite unexpectedly, five years ago I took a path full of incertitudes. Looking back, I realize that some persons made the journey along this path possible.

If I would have to choose the e-mail which had the biggest impact on my career and life this would be the one in which Alessandra confirmed me that I was accepted to the bioinformatics master in Paris. In the last years I had the chance to be in Alessandra's group. Her passion to make science happen and her continuous support to the team inspired me the type of scientist I want to become. After all these years, I have finally the chance to thank her for all the advice she gave me and for helping me establish a direction in my career.

My initial approach to science was dominated by the "Monte Carlo spirit". I was constantly sampling ideas and constantly ignoring any kind of "acceptance probability". I would like to warmly thank Hugues and Pierre for helping me concentrate my ideas. It must have been a difficult task for them, but I can assure them that their efforts paid off. Thank you Hugues for your continuous support, for remaining optimist even when results were disastrous, for coaching me along my teaching classes and for a showing me around Berlin. Pierre, among all the things I learned from you, one is in particular important for me. I knew that the word "exact" exists but it seems I was ignoring its importance. Thank you for your patience, for showing me the beauty of this word, for your open spirit and for your constant concern.

I am honoured that Christophe Ambroise, Jean-Philippe Vert, Nicolas Chopin , Claude Thermes and Jean-Daniel Zucker accepted to take part in the jury of my Phd defence. The questions you have raised on the topic of my dissertation, and the ones to come during the defence, challenged me to deepen my research. I thank again Claude Thermes, together with Adeline Samson and Martin Weight, for their guidance after participating at the mid-term evaluation of my thesis and for their suggestions on how to improve my research.

If today I am able to affirm that my doctoral studies were one of the most interesting periods of my life, it is also because of the people I met in the "Génomique des Microorganismes" and "Mathématique, Informatique et Génome" laboratories. I well remember the morning coffees with Claire which transformed into long discussions about the science of philosophy. With Vittore and Raphael I have discovered that even confusing questions may have meaningful answers. Thank you Vittore for digressing with me on unprobable subjects like scientific democracy, evolution of monogamy and, for allowing me to cook you pasta. I acknowledge every of my colleagues from GM (now BCQ) and MIG with whom I've shared so many interesting moments. Thank you Cyprien for sharing your office with me along these years, thank you Gilles for the first lecture I had on genomics. It would take a full page to mention all the beautiful persons I was given the privilege to meet and to learn from along these years. I will surely miss the human diversity and the positive work environment I have benefited off during my thesis.

I have been blessed with an amazing family. I thank my parents for always supporting me and my brother for allowing me to dream about translating the HMM into art paintings.

My wife, Adina, and our son, Eliad, bring beauty in my life every day and give me extra strength to confront the ups and downs in my work. I thank Adina for challenging me to rethink the artificial limits tradition sometimes imposes. I also thank her for the "suggested" readings on my nightstand, for her interest in my work and for her patience when our discussions became intricate.

Contents

List of Figures

List of Algorithms

List of Tables

General Introduction

In this thesis we address the problem of reconstructing the transcription profile from RNA-Seq reads in cases where the reference genome is available but without making use of existing annotation. An important result of this work consists in the design of a new model for the RNA-Seq read counts. We also extend the analysis to account for multiple conditions and notably design a method to estimate regions with difference in expression (DE) without using previously defined transcription units.

The first chapter consists of an introduction to the biological context and high-throughput sequencing where particular attention is paid to presenting the transcriptome investigation by RNA-Seq. RNA-Seq results in the determination of the sequences of transcript fragments (reads). Mapping the reads to genomic positions reveals the regions that are transcribed. The distribution of reads mapped to positions within such a region should present some property of homogeneity. Based on this, we expect that significant changes in read coverage along the genome can point to transcription breakpoints.

The second chapter deals with statistical methods that can be used in the analysis of series of counts. In the framework of State Space Models (SSM), models that pertain to estimating latent trajectories from series of observations, we present current methods for parameter and latent trajectory reconstruction. The Sequential Monte Carlo (SMC) methods are particularly fit for inference in complex SSMs. We focus on presenting Particle Monte Carlo Markov Chain and in particular the Particle Gibbs, a Monte Carlo Markov Chain algorithm that uses SMC updates of the latent profile as a proposal.

In the third chapter we present our contribution for the RNA-Seq read count model, the inference transcription profile by using Particle Gibbs and the reconstruction of DE regions. Our initial work used Negative Binomial distributions to model the read count emission given the expression level (hidden trajectory). The analysis of several data-sets proved that this model is not generally valid. To address this issue we develop a mechanistic model which accounts for the randomness generated within all RNA-Seq protocol steps. Such a model is particularly important for the assessment of the credibility intervals associated with the transcription level and coverage changes. Parameter values within this model can be further used to describe protocol characteristics.

In the fourth chapter we describe the transcription profile reconstruction. We describe a SSM accounting for the read count profile for observations and transcription profile for the latent variable. For the transition kernel we design a mixture model combining the possibility of making, between two adjacent positions, no move, a drift move or a shift move. Then we detail our approach for the reconstruction of the transcription profile and the estimation of parameters using the Particle Gibbs algorithm. We analyse the accuracy of our Particle Gibbs implementation and compare it to the Sequential Importance Resampling algorithm. We then evaluate the results obtained in breakpoint identification and compare them to those obtained with other methods. This work was published in a Bioinformatics paper "Parseq: reconstruction of microbial transcription landscape from RNA-Seq read counts using state-space models" [Mirauta et al., 2014].

In the fifth chapter we complete the results by presenting an approach for analysing differences in expression without making use of existing annotation. The proposed method first approximates these differences for each base-pair and then aggregates continuous DE regions. We published this work in the proceedings of the ICIAP 2013 conference [Mirauta et al., 2013].

In the closing chapter of our thesis we try to summarize our observations on the genome wide analysis in a SSM framework using SMC methods and discuss possible extensions of this work. We also point out the perspectives of developing RNA-Seq evaluation criteria based on the the new parametrisation of the read count model.

Chapter 1

The Biological Context

We introduce in this chapter the biological context of this thesis on transcriptome reconstruction using high-throughput sequencing and notably RNA-Seq data. We describe here: 1) the main mechanisms of transcription, 2) the transcriptome sequencing protocols, 3) the main characteristics of RNA-seq data, and 4) current methods in transcriptome reconstruction from RNA-Seq data.

All cellular organisms synthesize molecules with similar biological processes. The DeoxyriboNucleic Acid (DNA) molecules encode genetic information and influence through subsequent processing the organisms' phenotype. The processing steps are described in what is known as the central dogma of molecular biology which states that the flow of genetic information within a biological system goes through three distinct steps. The first step consists in the replication of the DNA. The second consists in the transcription of DNA information into an intermediary molecule, the RiboNucleic Acid (RNA) molecules. Following transcription, the translation process uses specific processed RNA molecules to encode a different class of macromolecules: the proteins. Some RNA molecules perform cellular functions without undergoing translation.

Detecting the regions that are transcribed into RNA, and the rate of transcription in different conditions represents an important step in understanding the cellular activity. We will focus our discussion on RNA molecules and, in the first part of this chapter we will describe several RNA types and present the general mechanisms of RNA synthesis and post processing. Post transcriptional RNA regulation is of interest for our work in the measure it affects the nature and levels of the RNA in the different cellular compartments.

After a short digression into the history of transcriptome analysis we present the recent technologies, with a focus on the sequencing of RNA. A crucial step in the evolution of RNA analysis was the development of a method for the determination of the sequence of nucleic acids within a transcript. Hybridisation arrays and sequencing protocols allow a fast, accurate and quantitative detection of transcribed regions. We review a few popular transcriptome sequencing protocols and finally detail the RNA-Seq protocol. We give a canonical representation of the complete flow of the RNA-Seq protocol: from RNA selection to the mapping of fragments to DNA regions. This representation is important for our understanding of RNA-Seq data and we will refer to it several times in the next chapters.

Then, we detail the characteristics of sequences issuing from RNA-Seq protocols. Protocols and technology can have a significant influence and can introduce several types of biases. We describe the different sources of bias which were identified.

To end this chapter, we summarize the questions addressed in our analysis and notably the estimation of transcription breakpoints and transcription levels, and the detection of regions that have a different expression between two conditions. We present a few popular methods for the reconstruction of transcript boundaries and analysis of differential expression.

1.1 The Transcriptome

1.1.1 A Brief Introduction to Molecular Biology

Molecular biology finds its origins in the 19th century following advancements in microscopy and biochemistry and started to be an intensive research field during the scientific exuberance of the early 20th century. Early research had a focus on the isolation and classification of macromolecules and resulted in the identification of the main biological polymers: DNA, RNA and proteins. Further on, biochemistry led to the identification of molecular differences between DNA and RNA [Frank, 1941]. Between the 1950's and the 1960's three major discoveries were made: the determination of the double helix structure of the DNA, the central dogma of molecular biology, and the genetic code that explains protein encoding in nucleotide sequence. The second one, restated in [Crick, 1970], refers to a hypothesis (that since became a paradigm) which states the relation between DNA, RNA and proteins and thus allows shaping the description of the molecular landscape of the cell.

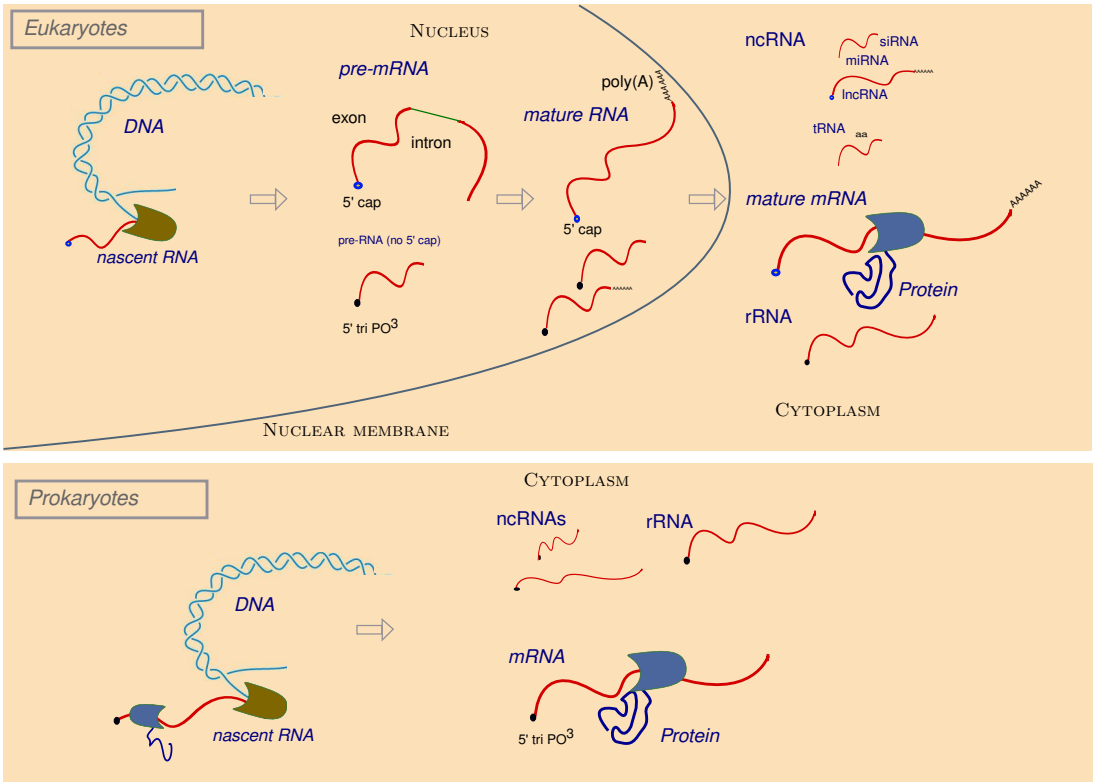

Figure 1.1: Transcription of DNA into RNA and translation of RNA into proteins. We show most frequent steps for eukaryotes (top) and prokaryotes (bottom).

It is now well established that protein synthesis in a cell is done in a two-step process:

- Transcription: genetic information is copied from the DNA into an intermediary molecule (RNA).

- Translation: protein polymers are synthesized using the messenger RNA (mRNA) template. In prokaryotes, the translation is coupled to transcription in the sense that it can occur co-transcriptionally. In eukaryotes, the translation uses mature mRNAs and takes place in the cytoplasm (after the export of mRNA to the cytoplasm).

1.1.2 The RNA Molecules

The RNA is a heteropolymer, a chain of different types of molecules. The monomers (named nucleotides) contain a ribose sugar (with carbons numbered from 1' to 5'), a phosphate group and one out of the 4 bases: Adenine, Cytosine, Guanine and Uracil (A, C, G and U). The nucleotides are covalently (sharing of electron pairs) bonded though a phosphate group attached to the 3' position of one ribose and the 5' position of the next. To refer to one monomer we will use both nucleotides (nt) and basepairs (bp) in reference to the building blocks of the DNA double helix. RNA is composed of nucleotides closely related to those found in the DNA with uracil replacing thymine. Unlike DNA, the RNA has the ribose sugar instead of the deoxyribose making the bonds less resistant and it is in general single stranded. Typical RNA lengths range between 10^2 and 10^4 nucleotides (10^6 to 10^9 for a typical genome length).

We can identify three classes of RNA given their transcription and post transcription processing status:

1. nascent RNA. These RNAs are in the course of being transcribed and are located in the nucleus.

2. precursor RNA (pre-RNA). These RNAs have been transcribed and currently undergo post-transcriptional processing in the nucleus.

3. mature RNA. These are RNAs that underwent post transcriptional processing. The mature RNAs can be located in either the nucleus or in the cytoplasm.

The RNAs play a wide range of roles in the cell. In the sense of the central dogma of molecular biology, the RNA serves as a template for protein synthesis (translation). The RNAs that play this role belong to the class of messenger RNAs (mRNA).

A heterogeneous group of RNAs are involved in mRNA regulation (tuning of the amount of specific mRNA molecules within a cell). In eukaryotes several types of non coding RNAs (ncRNA) are known to repress transcription, impede RNA translation or promote degradation. Some of the best known are two small RNAs: the microRNAs (miRNA) (single strand 21-22 nt) and small interfering RNAs (siRNA) (double strand 20-25 nt). The miRNA and siRNA regulate mRNA by controlling its interaction with a protein complex (RNA-induced silencing complex - RISC) resulting in translation blocking and respectively mRNA cleavage (reviewed in [Lee et al., 2004]). Other ncRNAs, the long noncoding RNAs (lncRNA) - (200+ nt), are involved in various mechanisms of regulation (reviewed in [Rinn and Chang, 2012]).

Other classes of RNAs have functions in the translation process. The small nuclear RNA (snRNA) - (150 nt) are involved in the pre-RNA splicing. The transfer RNA (tRNA) - small (80 nt) transfers associated amino acids to the ribosome according to a three-nucleotide sequence complementarity requirements (the genetic code). The ribosomal RNA (rRNA) - long (1000+ nt) is a component of the ribosome (the translation machinery).

1.1.3 The RNA Synthesis: Transcription

The RNA synthesis is a complex process that can be divided in several steps: initiation, elongation and termination. During **initiation**, an enzyme known as the RNA polymerase (RNApol) attaches, through the mediation of several associated molecules, to the template DNA strand.

In prokaryotes the sigma factor is the subunit of the RNApol that recognizes the promoter, a region upstream the Transcription Starting Site (TSS) (figure 1.2). There are two main structurally unrelated families: σ^{70} (most frequent) and the σ^{54} (reviewed in [Kazmierczak et al., 2005]). Sigma factor binds sites (SFBS) are usually organised in two boxes positioned at approximatively -35 and -10 nt upstream of the TSS that show some level of sequence consensus. For illustration, in the σ^{70} family, the σ^{70} subfamily binds to promoters presenting sequences related to the consensus TTGACA at -35 and TATAAT

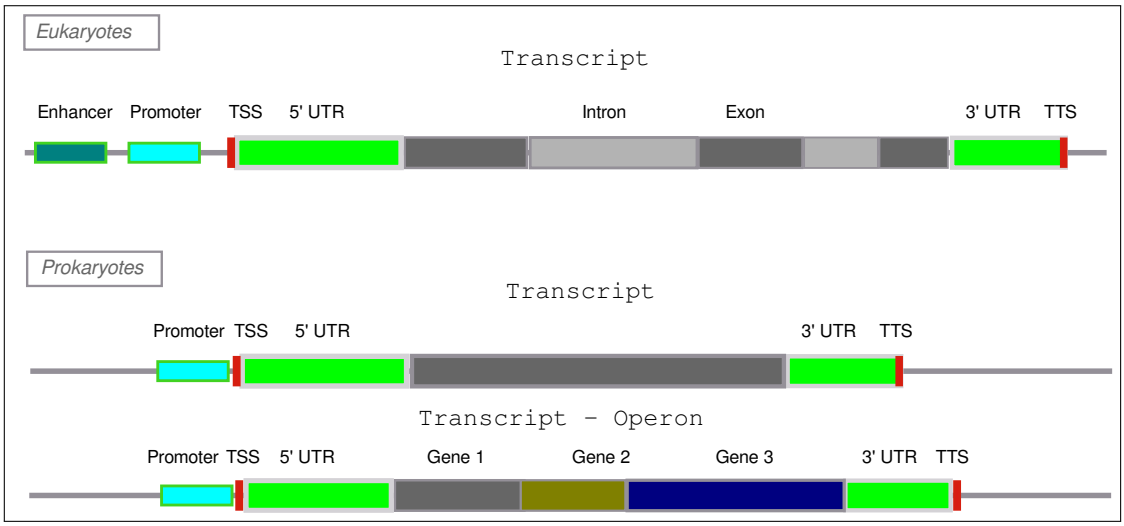

Figure 1.2: Transcription in eukaryotes and prokaryotes. A typical transcriptional unit is composed from translated regions (exons) and untranslated regions (introns and UTRs). The boundaries are named Transcription Start Site and Termination Site (TSS and TTS). We show also the regions where the transcription factors mediate RNA polymerase binding (enhancer and promoter).

(the Pribnow box) at -10 [Schaller et al., 1975]; the σ^{32} subfamily recognizes the consensus CTTGA at -35 and GNCCCCATNT at -10 (*E.coli*, [Wang and deHaseth, 2003]), where N denotes any nucleotide. The sigma factors of the family σ^{54} shows similarly a consensus sequence between -26 and -10 nt upstream TSS [Francke et al., 2011].

Three different types of RNA polymerases coexist in eukaryotic cells (RNApol I for rRNAs, RNApol II for mRNA and RNApol III for ncRNA). In this case, the RNAPol binds through a more complex process usually mediated by several transcription factors and enhanced by other regions (enhancers). There is a wide range of consensus forms for transcription factors binding sites (e.g. the TATA boxes -25 bp upstream for RNApol II) [Yang et al., 2007].

RNA polymerase leaves the promoter (**promoter clearance**) to start transcription elongation. Factors impeding clearance (like the affinity of promoter-RNApol bonds) may induce the release of the nascent transcripts (2 to 15 nt. long), a process known as abortive initiation [Goldman et al., 2009].

During **elongation** the RNApol creates a copy of the DNA coding strand (the complement of the template strand) by base pairing free nucleotides to the template DNA

strand. The RNA molecule is extended from 5' to 3' (where 5' denotes the 5th carbon of the ribose backbone). The elongation process can be paused and the same gene can be transcribed simultaneously by several RNAPols. Transcription takes place with a rate of several dozens nucleotides per second (for RNA pol II it has been estimated at $18 - 42$ nt. per second [Prez-Ortn et al., 2007]).

Transcription **termination** occurs at sites called Transcription Termination Sites (TTS) after the RNA pol receives termination signals (related mostly to sequence patterns):

- Prokaryotes: specific sequences that foster loop formation and weak poly(A)-poly(U) structures and thus destabilize the RNA-DNA hybrid are a sign for transcript termination [Carafa et al., 1990]. Termination can also involve additional molecules that interact with the RNAPol and force it to detach from the DNA (e.g. binding of the Rho factor to the mRNA [Richardson, 2002]).

- Eukaryotes: termination processes depended on the type of RNAPol (reviewed in [Richard and Manley, 2009]). For RNAPol I termination requires a specific DNA-binding termination factor (unlike the prokaryotic Rho factor which binds to the RNA). RNAPol II termination is coupled with 3'-end processing and may occur between a few base pairs to several kilobases downstream from the 3'-end of the mature RNA. RNAPol III terminates after a poly(A) sequence (not requiring previous RNA loop structures).

Processing of precursor transcripts (pre-RNA) differs greatly between the domains of life. In general terms the processing includes boundary (5' and 3' ends) and internal changes.

- Prokaryotes: these changes are known to be less frequent. They include: 1) conversion of the triphosphate from the 5'-end of nascent RNAs (above 20nt) into a monophosphate, 2) synthesis of poly(A) tails for some transcripts (that can promote degradation).

- Eukaryotes have a more complex processing (explained in more details in [Lodish et al., 2000]). Common transcript changes include:

- 5' capping. Immediately after transcription initiation the 5'-end is processed: 1) the terminal phosphate group is removed; 2) a guanine is added to the end, 3) the guanine is 7-methylated and 4) eventually the following two bases are also methylated. Small variations exist between RNA families.
- Template-independent addition of As at the 3'-end, in a process called polyadenylation. Most mRNAs and some ncRNAs (like lncRNA) have poly(A) tails. The preRNA is first cleaved by an endonucleases that recognizes conserved motifs like AUUAAA at 10 -35 nucleotides upstream the cleavage site. After this, poly(A) polymerase synthesizes the poly(A) tail. The polyadenylation is important for the export of the mature RNA from the nucleus and the translation.
- Splicing [Berget et al., 1977; William Roy and Gilbert, 2006] is the RNA modification step that consists in removing non-coding sequences called introns and joining the protein-coding sequences called exons. A given pre-mRNA molecule can be spliced at different junctions (alternative splicing) to result in a variety of mature mRNA molecules, each containing different combinations of exons. These combinations are called isoforms.
- Editing is a relatively rare process which consists in changes to the RNA sequence.

Because eukaryotic transcription and translation is done in different compartments, the mRNAs must be exported (**transported**) from the nucleus to the cytoplasm through the nuclear pores.

After a certain amount of time the RNA **degrades** into its component nucleotides with the assistance of ribonucleases. In prokaryotes this process involves degradation of 5'-and 3' ends by exonucleases and internal cleavage by endonuclease (a more detailed description in [Evguenieva-Hackenberg and Klug, 2011]) and typically occurs after a few minutes. In eukaryotes the degradation is done mainly by exonucleases through shortening of poly(A) tails and de-capping. The mRNA half lives tends to be longer in eukaryotes (it is up to several hours whereas the synthesis (transcription) takes less than a minute for a typical transcript).

1.2 High-Throughput Sequencing

The RNA and DNA molecules have a natural sequential organization enforced by covalent bonds. Beside detecting macromolecule composition [Frank, 1941], a step forward in their analysis consisted in the identification of the arrangement of composing subunits (nucleotides). In genomics, the term **sequencing** pertains to methods that aim at determining the sequence of monomers in polymers (nucleotides in the case of DNA and RNA). The sequencing methods build upon physicochemical properties of DNA and RNA. They consist in the reconstruction of the complementary strand with the incorporation of labelled nucleotides or use complementarity to associate molecules to known sequences through hybridization. The first sequencing methods concerned the determination of peptide amino-acid sequences through partial hydrolysis. They were performed by Sanger in 49-51 and were followed by several methods to determine the DNA sequence (among which Wu [1972] using synthetic labelled oligonucleotides). After 1990, with the advent of sequencing by chain termination and the development of DNA arrays, genome wide sequencing became possible. We describe here shortly the Sanger chain termination method, the sequencing by ligation and focus on the sequencing by synthesis:

- **Sequencing by chain termination** (Sanger). The first approaches [Sanger et al., 1977] consisted in two steps: 1) polymer synthesis by incorporation of common deoxynucleosidetriphosphates (dNTPs) and of modified dNTPs that terminate DNA strand elongation and 2) size selection. Sequences obtained after addition of modified dNTPs of only one type (only Adenine for example) and selected for the size k (by electrophoresis) provide information about the presence of the nucleotide (Adenine in the example) at position k. In this protocol 4 different mixtures are required (one for each dNTP) to determine the complete sequence. Current Sanger protocols use fluorescent dyes to distinguish the bases in the same electrophoresis.

- **Sequencing by synthesis** (Solexa-Illumina). Sequencing by synthesis reveals the sequence of a DNA polymer during the synthesis of its complementary strand enzymatically with marked (fluorescent) nucleotides. In current Solexa-Illumina protocols sequencing requires several preparation steps and is done in a multi-cycle process but synthesis is done in parallel for the entire library. The DNA material is usually fragmented and fragments within a given range are selected. Then, adaptors

are ligated at one or both ends and the fragments are bound to a flowcell (a surface with separately contained lanes). To increase signal strength, each fragment is amplified into a clonal cluster with approximatively a thousand copies of one original molecule fragment.

The multi-cycle process is started to construct and determine step by step the complementary strand of each molecule in each cluster. During a cycle, several actions are performed for all clusters: 1) labelled nucleotides (incorporating additional molecules that stop elongation) are added to the flowcell ; 2) nucleotides are incorporated (by DNA polymerase) in the growing strand (the complement of the template sequence); 3) images of the flowcell are capture allowing the detection of dominant labels (corresponding to a nucleotide) for each cluster and 4) free nucleotides, blocking molecules, and fluorescent groups are removed to allow for a new cycle. At each step the sequence corresponding to each cluster is obtained by assessing the nucleotide dominantly incorporated. The blocking molecules impede the addition of two nucleotides to the same sequence in the same cycle.

- **Sequencing by ligation (SOLiD)** extends the growing chain by ligating oligonucleotides (instead of using polymerases to add single nucleotides like in sequencing by synthesize). It is done in two steps: 1) labelled oligos are hybridized to the target (which is single stranded) and 2) the 5' of the growing strand is ligated (by using ligase) with the 3' of the oligo. The result consists in a label (colour) sequence that is afterwards used to infer the nucleotide sequence. Issues were reported concerning palindromic and repeated regions [Huang et al., 2012].

While the initial Sanger protocol determined one sequence at a time, the Illumina and SOLiD are high-throughput protocols, i.e. they process in parallel tens of millions of sequences. Currently, Illumina second generation platforms are dominating the sequencing industry. According to the information on www.illumina.com and www.appliedbiosystems.com websites, both Illumina and SOLiD systems are comparable in terms of sequencing costs and throughput: 25 Gb of mappable reads per day at 70euro per Gb for the Illumina HiSeq2000 and 6 Gb of mappable reads per day at 45 euro per Gb for SOLiD (as of September 2014). Read length ranges between 30 bp (early Illumina) and several hundreds (SOLiD). Each run produces 100 million+ reads. Both Illumina and SOLiD can

sequence one end of the fragments (single-end reads) or both (paired-end or mate pairs). Paired-end reads represent pairs of reads where each one was obtained through sequencing of one fragment end. The region between the pair of reads is called insert. Mate pair reads have a longer insert size and in this case the fragments are circularized and the regions containing both ends (which are now connected) are sequenced. Longer sequences (above 20kbp) can be obtained using third generation sequencing protocols (e.g. Helicos Bioscience and Pacific Biosciences).

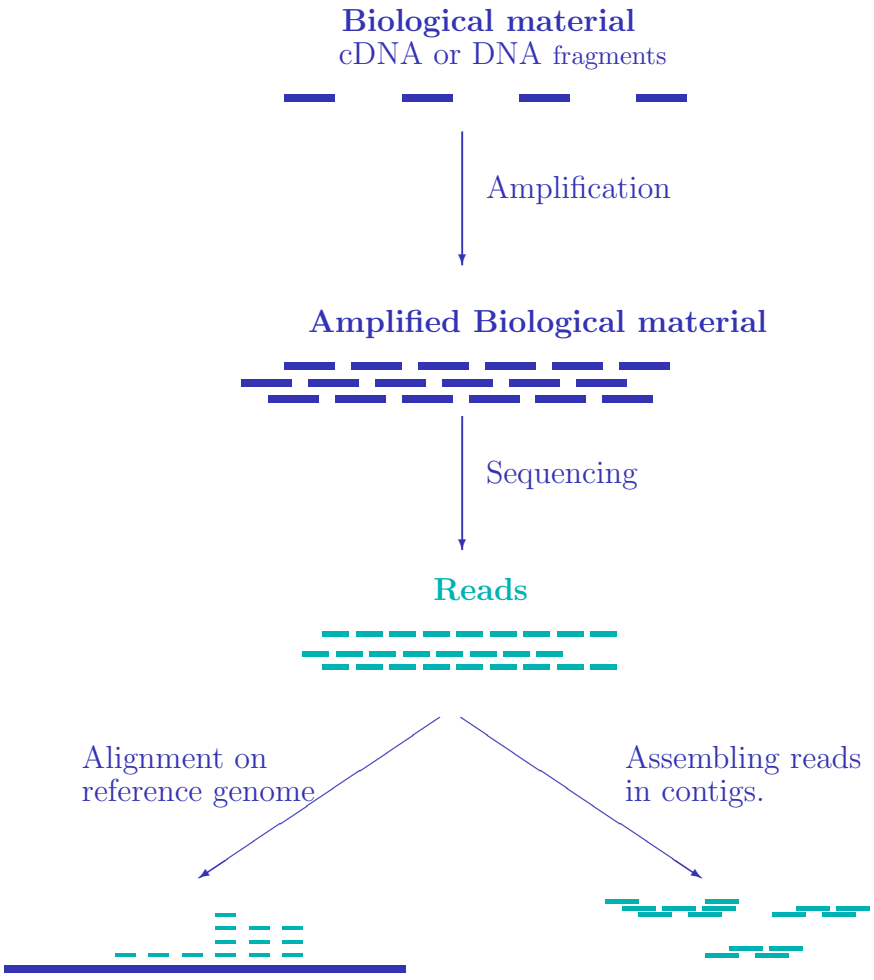

Figure 1.3: Canonical steps for sequencing the DNA. The biological material (DNA from DNA-protein complexes, cDNA obtained from RT of RNA, DNA fragments, etc.) is amplified to reach a sufficient amount for sequencing. Sequencing is performed on the amplified molecules in a multi-cycle process on full fragments or only fragment ends. The reads are then aligned to a reference genome or assembled into contiguous longer segments to reconstruct original molecules.

There is a wide range of DNA sequencing protocols but in principle they follow the canonical steps depicted in figure 1.3. The biological material, and thus the biological questions to be answered, drive several approaches to library construction. We briefly discuss some of the most interesting protocols for selecting the biological material:

- **Chip-Seq** [Barski et al., 2007] aims to identify the binding sites of DNA-associated proteins. The biological material is selected through immunoprecipitation (ChIP) of DNA-protein complexes after covalent cross-linking of DNA and proteins. Sequencing is performed on the DNA regions detached from these complexes.

- **3C-Seq** [Dekker et al., 2002] or Chromosome Conformation Capture aims to capture DNA regions that interact directly or indirectly. The biological material is obtained by cross-linking interacting regions on the genome and subsequent digestion of non cross linked segments.

- **BS-Seq** (or Bi-Seq) [Krueger et al., 2012] aims to identify methylated positions (cytosines), i.e. positions that have and additional CH_3 group. Non-methylated Cytosines from DNA molecules treated with bisulfite change into uracil. The genomic position of BS-Seq reads can be determined by aligning the non-C nucleotides. By examining at the genomic positions of the cytosines the corresponding nucleotide found in the reads one can estimate the percentage of methylation.

- **Tn-Seq** [van Opijnen et al., 2009] aims to analyse the fitness of specific genomic changes. A marker is inserted in a random manner by a transposon into the genomes of an organism. The same population is sequenced after a period of selection. Sequencing regions surrounding the marker allows the determination of positions where the insertion has a positive or negative impact on fitness (typically by gene inactivation).

1.3 Methods for Transcriptome Investigation

Currently, the transcriptome investigation is done through array hybridisation or high-throughput sequencing.

Taking advantage of hybridisation properties of single strand DNA, **the array based technologies** can be used for investigating the transcriptome (Microarrays [Schena et al., 1995] and Tiling arays [Bertone et al., 2004]). In a typical array experiment, a population of transcripts are first reverse transcribed (RT) and marked with fluorescent dyes or biotin. Then the marked cDNA molecules are hybridised on an array. A DNA microarray consists in a collection (10^3 to 10^5) of DNA probes (of known sequence) attached to a solid surface. After washing, the molecules that contain sequences complementary to the attached probes will remain on the array. Subsequent scanning and analysis of the light intensity allows the determination and quantification of the cDNA and therefore RNA molecules that are present in the library. Contrary to the classic microarrays, the tiling arrays include genome-wide selected probes. It permits the quantification of the expression level at each position present in the probes without prior information on the transcribed regions.

As for hybridisation based methods, **the sequencing of RNA molecules** relies on the reverse transcription of RNAs into cDNA. The ribonucleic chain is less resistant to hydrolysis than the deoxyribonucleic chain (due to its additional hydroxyl group). Recent technological developments lead allow direct RNA sequencing [Ozsolak et al., 2009] but these protocols are not yet applied on a large scale. Including the RT and post sequencing steps that assign a read to its genomic position, a canonical RNA sequencing protocol involves the steps presented in figure 1.4.

The RNA-Seq [Mortazavi et al., 2008; Wang et al., 2009] aims to profile the complete transcriptome without prior knowledge of the transcribed regions. The procedure starts with the isolation of the RNA molecules of interest. After DNA depletion common RNA libraries may target different populations of RNAs such as poly(A) RNA (selected by binding the molecules on poly(T) beads), or small ncRNA (isolated by size selection). Other techniques include transcripts involved in RNQ -protein complexes like Ribosome profiling (RibSeq) for transcripts active in translation or Global run-on-sequencing (GRO-seq) for nascent transcripts. RNA molecules are usually further fragmented. The fragments

are Reverse Transcribed (RT) into cDNA molecules. In most current protocols, before RT, strand labelling and degradation of RT unmarked cDNA products allow to maintain strand information. A size selection step may be performed before or after RT. The library preparation includes also end repairing and adapter ligation steps. After library preparation, sequencing may be performed using standard techniques.

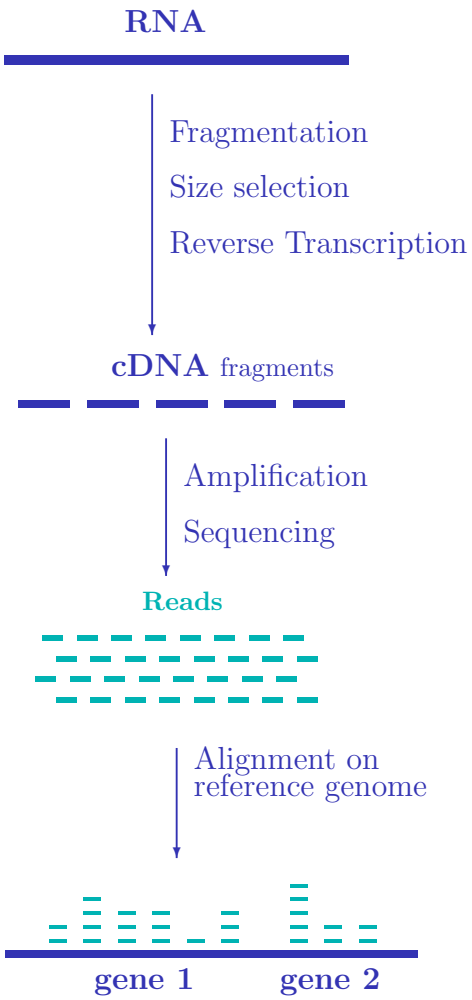

Figure 1.4: Canonical RNA-Seq flow protocol steps. Beside the steps presented in figure 1.3 the RNA-Seq includes RT of RNA into DNA. Strand specificity of reads aligned to the genome is assured at this step.

For protocols aiming at sequencing transcript boundaries the libraries are enriched in molecules containing the transcripts 5'-ends or 3'-ends. Cap analysis of gene expression

(CAGE) [Shiraki et al., 2003; Takahashi et al., 2012] and differential RNA-Seq (dRNA-Seq) [Borries et al., 2012] are two such methods that aim to sequence 5' ends of transcripts:

- Cap analysis of gene expression (CAGE) and its unamplified HeliscopeCAGE version [Kanamori-Katayama et al., 2011] may be applied to sequence transcripts with 5' cap (eukaryotic mRNAs). After capturing of 5'-ends, fragments that have the exact same length obtained with restriction enzymes are joined (with the corresponding adaptors) in a continuous vector and this vector is then sequenced. The resulting tags are counted providing quantitative evaluation of the frequency of using a specific TSS. The CAGE protocol was built upon Serial analysis gene expression protocol (SAGE) [Velculescu et al., 1995].

- Differential RNA-Seq (dRNA-Seq) is used for the analysis of primary transcripts in bacteria. It aims at distinguishing 5'-ends of the primary transcripts (bearing a 5' triphosphate) from those generated through RNA processing and degradation (bearing a 5' monophosphate). To this end it creates two libraries from which one is enriched in primary transcripts using monophosphate sensitive exonucleases (TEX - terminator exonuclease or TAP - tobacco acid phosphatase). Detection of primary transcripts 5'-ends is achieved through a differential analysis of the two libraries.

1.4 RNA-Seq Data

RNA-seq reads represent transcript fragments and thus, when aligned to a reference genome, should map relatively uniform within transcribed regions. In figure 1.5 we show an example of reads aligned to the genome for *C.albicans*. A simple visual analysis of this example spotlights some issues with respect to the read coverage. First, we observe reads that map outside known (probable) transcription units. These might correspond to unknown genuine transcripts or to a background noise. Second, within a transcript the read coverage is not very uniform and more, we observe frequently coverage gaps.

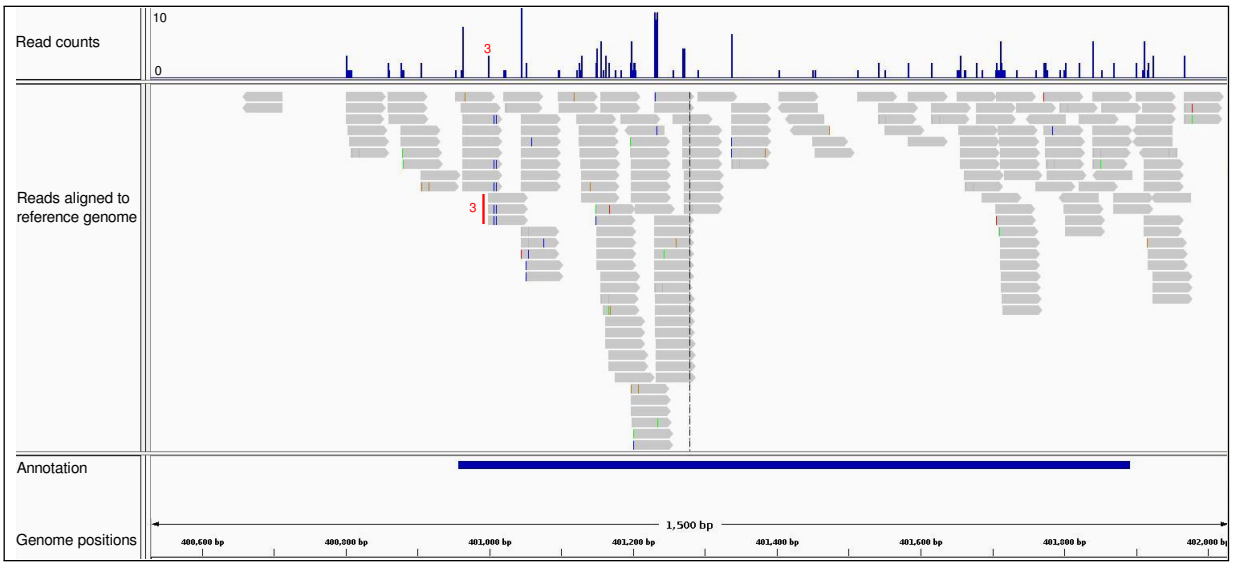

Figure 1.5: Integrative Genomics Viewer (IGV) [Thorvaldsdottir et al., 2013] screen shot of RNA-Seq reads aligned on the genome of *C.albicans*. Bottom lane: genomic positions. Middle lane: reads aligned on the genome represented by directed blocks. The direction of the block represents the read sense comparing to the reference strand. Top lane: read counts. We count the number of reads with the 5' end at a genomic position. For illustration, there are 3 reads (corresponding to the red bar) with the 5'end mapping at position 401k.

The canonical RNA-Seq protocol (Figure 1.4) has several steps that induce randomness of the results and possibly biases. In the most simple scenario we expect that the fragmentation, amplification and selection of reads are done homogeneously along a transcript and that read coverage shows only variability issued from independent identically distributed fragment sampling processes along the sequence. Real datasets prove that read coverage is affected by other factors and that several forms of bias exist. The high

dependency of the data on the protocol conditions [Khrameeva and Gelfand, 2012] makes it difficult to trace back at which step of the protocol the biases occur and in the same time impede drawing general conclusions. Thus, we shall limit ourselves to describe some observations on biases that are recurrently mentioned in literature and we will focus on those observed in Illumina protocols. In the Results chapter 3, we discuss bias observed in several bacterial and simple eukaryotic datasets.

Read counts may be influenced by the transcript sequence, the position of the read 5'-end relative to transcription boundaries and the transcript conformation.

Regions with elevated **GC content** tend to have a lower read coverage [Benjamini and Speed, 2012]. One explanation [Aird et al., 2011] is related to the PCR amplification that drops down for both very low (10%) and high (70-90%) local GC content (the later only for high heating rates). This might be caused by poor denaturation due to strong GC pairing, poor access of the PCR primer and slow elongation due to template secondary structure.

Beyond the GC content, **the genomic sequence surrounding the read 5'-end** may also affect the uniformity of read distribution [Li et al., 2010; Hansen et al., 2010]. This influence is mediated by the formation of structures that impede primer binding in RT, the preference of primer non-random flanking tags to specific RNA sequences, or PCR amplification.

Besides the RNA sequence the biases may be induced by other factors. The **conformation of RNAs** which is partially induced by the sequence and partially by the environmental conditions is thought to influence the count distributions [Li et al., 2010; Jackson et al., 2014]. Secondary and tertiary transcript structures might impede random fragmentation, reverse transcription and amplification.

Count distribution within a transcript might be affected by the **position relative to transcript boundaries**. Results concerning this bias show a wide range of behaviours. In Bohnert and Ratsch [2010] it was observed that for a *C. elegans* dataset (poly(A) enriched) the read coverage is higher close to the 5' end and lower close to the 3' end. On the other hand [Wan et al., 2012] and references within conclude that RNA degradation affects read coverage in a different way: counts are expected to be higher at the 3' ends when RT is done on poly(A) RNAs.

It has also been found that **palindromic sequences** are under-represented in se-

quencing by ligation results. A hypothesis to explain this bias invokes the possibility of forming hairpin structures that could impede oligo-sequence hybridization [Huang et al., 2012].

Artefacts may also be related to sequencing, read mapping (notably for low-complexity and almost repeated regions) and DNA contamination. Sequencing and alignment artefacts can be in general mitigated by increasing read lengths. As errors tend to accumulate at the read 3'-end, proper read quality trimming helps improving alignment results. **DNA contamination** might lead to the presence of reads in otherwise non transcribed regions.

In conclusion, read coverage seems to be influenced by several sources of bias. Most of them are related to the transcript sequence and might originate from any of the protocols steps: RT, fragmentation, amplification and sequencing. While biases have various manifestations, these tend to be alike for a same protocol in the same conditions.

1.5 Transcriptome Investigation with RNA-Seq

There is a wide range of questions that can be addressed by using RNA-Seq data. Its current use is done mainly on data from a single organism but RNA-Seq may also be applied in a pluri-organism context (metatranscriptomics) to elucidate the transcriptional activity of an ecosystem [Gilbert and Hughes, 2011; Leimena et al., 2013].

A first category of results that can be derived from the analysis of the genome wide read coverage pertain to **transcript identification**. The reads represent transcript fragments and, after mapping on the genome, provide information of the location of transcripts. If the reference genome sequence is not available, contigs assembled from overlapping reads can help identifying the transcripts even if their genomic origin remains concealed. A step further, the reads and in particular the junction reads that span splicing positions allow the **reconstruction of different isoforms** of a gene.

Second, the read coverage of a particular genomic region provides information on the proportion of the corresponding transcript in the library. The aggregation of count within the boundaries of a transcript allow the estimation of the relative **expression level** (intensity of transcription) up to isoform level.

Third, a traditional question is, starting from two or more controlled experiments, to identify the set of elements that exhibit **differential expression**. Answering this question is an important step towards formulating a biological hypothesis or for instance deriving disease biomarkers.

Fourth, the sequence of reads overlapping a genomic position provide information on RNA post transcription processing. Differences in nucleotides between the reads stack and a genomic position might reveal **RNA editing** [Park et al., 2012; Ramaswami et al., 2013]. This point is not addressed in this thesis.

1.5.1 Transcript Quantification

The number of RNA-Seq reads corresponding to a genomic region is related to the steady-state quantity of transcripts issued from that region (resulting from the equilibrium between transcription and degradation). Thus, the reads can be used for a quantitative description of the transcriptome. Once reads are assigned to transcripts (by alignment on a reference genome or by assembling in transcription contigs) quantification can be done

at unit or at position level. For a unit, the sum of reads within the unit's boundaries provides a measure for the transcription level. For a position t, counts can correspond either to the number of reads whose 5'-ends map at position t, i.e. reads starting at position t, or to the number of reads overlapping the position t, i.e. the presence of the genomic position t in the read population. The total number of reads in an RNA-Seq experiment is arbitrary scaled in several protocol steps and depends on: the quantity of initial RNA, the number of amplification cycles, the rate of fragmentation, the selection and the amount of sequencing. Therefore measurement of expression level (transcription intensity) needs to be done in a relative manner.

The "Reads Per Kb per Million reads" [Mortazavi et al., 2008] (RPKM) tries to approximate the molar concentration of an exon in the total transcript population. For a total number of mapped reads C (known also in the literature as coverage or sequencing depth), an exon g with boundaries $g_{5'}$ and $g_{3'}$, length $l_g = |g_{3'} - g_{5'} + 1|$ and C_g the number of reads mapped within its boundaries, we write: $RPKM_g = \frac{C_g \cdot 10^3/l_g}{C/10^6}$. This provides for a unit g the average reads per position C_g/l_g scaled by the sequencing depth.

The expression quantification at isoform level can be done using junction reads [Wang et al., 2008] or directly from the exon expression level [Richard et al., 2010]. For two units g_i and g_j of a gene g, the second option tests the difference in their expression x_{g_i} and x_{g_j} to conclude if the two were jointly transcribed or if they might originate also from different transcripts. The approach (algorithm CASI from Richard et al. [2010]) supposes a Poisson distribution for the total counts within an unit (mean equal to unit expectation). Further on, the authors optimise the proportion of isoforms in order to explain the exon counts (algorithm POEM).

1.5.2 Methods for Transcript Reconstruction

Detecting transcription boundaries from global RNA-Seq data raises interesting and computationally difficult problems. We will present in this section some popular approaches and notably we will focus on methods that use the reference genome to identify the position of the reads. Methods that do not use reference genome, i.e. *de novo* methods, assemble transcripts from overlapping reads. A large number of methods of both types are reviewed in Martin and Wang [2011].

Reference based methods [Trapnell et al., 2010; Guttman et al., 2010; McClure et al.,

2013] and our own [Mirauta et al., 2014] begin with the alignment of reads on the genome. In specific cases, existing genome annotation (including ORFs or CDSs) provides sufficient information for transcript identification. In such scenarios the annotation (usually limited to CDSs) needs to be extended to include the 5'-UTRs and 3'-UTRs and to approximate the transcription boundaries. To extend a transcript g, the reads from its neighbourhood are assigned probabilities to be issued from g, from a background noise or from adjacent transcripts. One option to do so is by building a model for the distribution of the count of reads [McClure et al., 2013]. Other options assign reads based on some distance. For example reads that overlap a transcript have a high probability to belong to that transcript.

Cufflinks [Trapnell et al., 2010] aims at constructing the smallest set of transcripts that "explains" the reads observed in a RNA-Seq experiment. Transcripts are assembled from the mapped fragments (single-end reads or joined paired-end reads) sorted by reference position. Fragment overlapping is examined to build a connection graph. The connected fragments are predicted to belong to the same transcript. Isoform structure can be inferred from the path of fragment contigs and expression levels can be estimated after allocation of the reads to the inferred transcripts.

While this assembly approach provides insightful results at a computationally affordable cost and can use reads overlapping exon-junctions as direct evidence for splicing [Wang et al., 2010], it has also some intrinsic limitations. The most obvious is that limited depth of sequencing combined with technical biases may cause gaps that lead to artificial splits in the transcript structure. Irrespective of the sequencing depth, this approach is also unable to point to overlapping transcripts caused by promoter multiplicity and incomplete termination. However, these two mechanisms contribute substantially to transcriptome complexity in organisms with compact genomes [Nicolas et al., 2012].

1.5.3 Methods for Differential Expression Analysis

The question of identifying units with a significant difference in expression between biological conditions translates, in statistical terms, into detecting significant changes of expression level, after accounting for the sources of experimental and biological variability. However, this traditional statistical standpoint does not consider the magnitude of the effect, and authors proposed to overcome this limitation by directly testing whether

fold change is above a given level [McCarthy and Smyth, 2009].

The analysis of Differential Expression (DE) usually starts from predefined units of possible change, such as genes, exons or transcript isoforms. In this case, after cumulating counts at unit level (see section 1.5.1), one can estimate the statistical significance of the differences [Anders and Huber, 2010; Robinson et al., 2010; Trapnell et al., 2012]. The variability of read counts within a unit observed when re-sequencing the same library has been described as almost compatible with a Poisson distribution [Marioni et al., 2008]. However, when compared between samples (or replicate libraries), it exhibits over-dispersion and the negative binomial distribution is often used to accommodate this behaviour. Replicating the data sets permits to account for the biological and technological variability and mitigates the incertitude on expression level estimation. Challenging problems arise when the units of potential DE are unknown. In this case, one faces the problem of having to simultaneously delineate the boundaries of the DE region and estimating the magnitude of change.

Approaches to tackle this problem group in two categories. The first category consists of estimating the transcript structures from the different datasets (used separately or jointly) and then applying DE detection methods on predefined units. The second category computes the DE profile from the read coverage in the two conditions and then segment this profile into DE regions. The first option is made possible by several methods cited in the previous subsection that deal with transcript reconstruction. In this case, the DE units are derived from the estimated transcript structure and DE changes within those units remain invisible. Inferring DE regions directly from the DE profile provide a more detailed view of DE landscape but may raise problems concerning the correspondence to previous annotation. Using multiple replicates, Frazee et al. [2014] estimates first the DE at single base-pair resolution and then reconstructs the DE regions .

When we consider more than one condition, the depth of sequencing (e.g. the total number of reads produced) will directly affect the transcript expression level. Under a perfectly controlled experiment, this level is expected to scale linearly with the depth. In such conditions scaling by the read coverage (like in RPKM) should provide a satisfactory solution. However, Bullard et al. [2010] highlighted that in most transcriptomes samples a small fraction of the genes makes up most of the molecular mass, and thus simple scaling by the total coverage ratio could lead to very unstable normalization. Similarly to the

strategies for microarray data normalization (compared in [Bolstad et al., 2003]) several options, including notably quantile normalisation, can be adopted for RNA-Seq data (see detailed comparison in [Dillies et al., 2013])

In the next chapters we will present methods and results pertaining to transcription breakpoint estimation from changes in read coverage given a known genome sequence. The expression level is jointly estimated with the breakpoints. More briefly we will address also the differential expression question.

Chapter 2

Statistical Methods for Transcription Profile Reconstruction

In this chapter we describe the statistical methods which are the underlying basis of our approach for genome-wide transcription profile reconstruction. While RNA-Seq read counts provide various ways to detect transcript boundaries, our aim in this study is to develop a principled strategy for the approximation of the expression levels and the estimation of transcription breakpoints from longitudinal changes in the read coverage. We decide to approach this task in a state space model (SSM) framework. SSMs are a class of models that describe the probabilistic dependence between observed measurements and a latent (hidden) state variable. Along this thesis we use $(y_t)_{t=1:T}$ (or $y_{1:T}$) notation for a sequence of size T of observations and $(x_t)_{t=1:T}$ (or $x_{1:T}$) for the hidden trajectory (path).

We introduce the general filtering and smoothing recursions used for SSM inference. In discrete state space or Gaussian (linear) models the path reconstruction admits exact solutions. In general though, the SSMs raise mathematical challenges and hidden path reconstruction requires Monte Carlo approximations to resolve computational obstacles. We present Sequential Monte Carlo (SMC) algorithms that provide practical solutions for filtering and smoothing. With a view to achieve exactness, we choose for transcription profile approximation a recent Particle Monte Carlo Markov Chain method coined Particle Gibbs. Without being exhaustive we mention also other methods for trajectory reconstruction and parameter estimation within Bayesian and Maximum Likelihood frameworks.

2.1 The Estimation of Transcription Boundaries

2.1.1 Change Point Detection

Estimating transcription boundaries can be considered a classical problem of change point (breakpoint) detection where the sequence of observations $(y_t)_{t=1:T}$ needs to be partitioned into homogeneous segments. Here T is the genomic sequence length and y_t the read counts observed at the position t.

The basic method aims at partitioning the sequence into a predefined number of segments. It involves the following components:

- K segments delimited by $K-1$ breakpoints $b_{1:K-1}$;
- $(\tilde{x}_k)_{k=1:K}$ parameters for the segments $(b_{k-1} : b_k)_{k=1:K}$, (b_0, b_K - genome extremities);
- a cost function $C(y_{1:T}; b_{1:K-1}, \tilde{x}_{1:K}) = \sum_{k=1:K} \sum_{t \in (b_{k-1}:b_k)} C(y_t; \tilde{x}_k)$.

The cost $C(y_{1:T}; b_{1:K-1}, x_{1:K})$ measures how the partition $b_{1:K-1}$ with segment parameters $\tilde{x}_{1:K}$ fit the observed data $y_{1:T}$. Of great interest is the cost of a partition with optimized segment parameters

$$C(y_{1:T}; b_{1:K-1}) = \sum_{k=1:K} \min_{\tilde{x}_k} \sum_{t \in (b_{k-1}:b_k)} C(y_t; \tilde{x}_k).$$

Popular cost functions include the square error $SS(y_t; \tilde{x}) = (y_t - \tilde{x})^2$ and the minus log likelihood $L(y_t; \tilde{x}) = \log \pi(y_t \mid \tilde{x})$.

There are $\binom{K-1}{T-2}$ ways of partitioning the genome, and solutions to this need to be built considering this high dimensionality. Several partitioning methods can provide efficient solutions (reviewed in [Killick et al., 2012]). A first method, the binary segmentation, reduces the search by iteratively estimating single change point segmentations. In this algorithm an additional breakpoint that minimizes the cost is added at each step.

Dynamic programming, a term coined by Belmann in 1954 in the "Theory of dynamic programming", pertains to exact methods that deal with complex (multi-stage) problems by breaking them down into simpler sub-problems. For the partitioning problem it optimizes at each step $C(y_{t_0:t_1}; k)$, the cost of the optimal partition with k segments for the $t_0 : t_1$ sequence. The recursion writes: $C(y_{1:t}; k) = \min_{k-1<s<t} (C(y_{1:s}; k-1) + C(y_{s+1:t}; 1))$.

The computational cost $O(KT^2)$ can be improved up to being linear in T with simple (limiting breakpoint search using prior information on breakpoint positions or segment length [Huber et al., 2006]) or more sophisticated pruning methods [Rigaill, 2010; Killick et al., 2012].

Applications of these change point detection methods on genomic questions include work on CHG data analysis [Picard et al., 2005] and transcriptome reconstruction with data from tiling arrays [Huber et al., 2006] (with square error cost function) or Next-Generation Sequencing [Cleynen et al., 2014b] (with Poisson and Negative Binomial likelihood cost). A description of methods for multivariate scenarios is given in [Lavielle and Teyssire, 2006; Picard et al., 2011].

2.1.2 State Space Models

The State Space Models (SSMs) are other commonly used tools for change-point detection. These allow flexible assumptions on the chain to be partitioned and can provide an extended range of results on longitudinal dynamics.

The SSMs are models aiming at the reconstruction of the latent states $x_{1:T}$, not directly available, from the observed measurements $y_{1:T}$. For this it uses probabilistic models to link observations to the latent space and to describe the latent space dynamics. The SSM we use incorporate single data emission densities $\pi(y_t \mid x_t) =: e(y_t; x_t)$ and a Markov model for the latent variable x_t with values in the state space Ω_x and governed by the transition kernel $\pi(x_t \mid x_{t-1}) =: k(x_t; x_{t-1})$. SSMs with discrete Ω_x state space are often, and in particular in the computational biology literature, referred to as Hidden Markov Models (HMM). SSM models might be built on various choices of the emission and transition models.

Mild assumptions on the transcription profile and on the count emission allow us to design a standard SSM (DAG in figure 2.1), this framework can accommodate more complicated scenarios for transcription dynamics and emission. The theoretical model we present could be applied also for the higher-order Markov chains and emission models where the counts y_t depend locally on the hidden path ($\pi(y_t \mid x_{t-k:t+k})$) or on the hidden path and previous observations ($\pi(y_t \mid x_{t-k:t+k}, y_{1:t-1})$). Solutions to these models may demand simple adjustments to the model densities and recurrence relations or may require state space augmentation (transforming the problem into a standard SSM with a higher

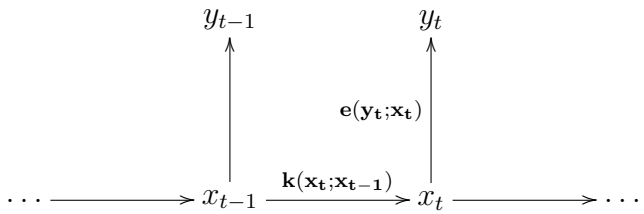

Figure 2.1: Direct Acyclic Graph (DAG) for a standard Space State Model. Variables: $(y_t)_{t\geq 1}$ observations and $(x_t)_{t\geq 1}$ hidden states. y_t depends only on x_t according to the $e(y_t; x_t)$ density; $\mathbf{x}$ is a first order Markov chain with transition density $k(x_t; x_{t-1})$.

dimension state space).

2.1.3 Sequential Hidden Path Reconstruction in SSM

In the context of general SSMs, the reconstruction of the hidden trajectory from the sequence of the observations, i.e. the analysis of $x_{1:T} \mid y_{1:T}$, is done in a sequential manner. It utilizes a decomposition of the complete density $\pi(x_{1:T} \mid y_{1:T})$ according to the DAG from figure 2.1) into "pieces" that can be expressed in terms of emission (e) and hidden transitions (k) densities. The decomposition writes

$$\begin{aligned} \pi(x_{1:T} \mid y_{1:T}) &= \frac{1}{\pi(y_{1:T})} \cdot e(y_1 \mid x_1) \cdot \pi(x_1) \cdot \prod_{t=2:T} e(y_t \mid x_t) \cdot k(x_t \mid x_{t-1}) \\ &\propto e(y_1 \mid x_1) \cdot \pi(x_1) \cdot \prod_{t=2:T} e(y_t \mid x_t) \cdot k(x_t \mid x_{t-1}), \end{aligned} \tag{2.1}$$

where the $\propto$ (proportional) symbol indicates equality up to a constant between two densities.

In the context of genomic analysis, we are generally interested in complete path reconstruction $\pi(x_{1:T} \mid y_{1:T})$ and in the marginal distribution $\pi(x_t \mid y_{1:T})$. For a position t there are several quantities of interest that can be combined in the recurrence relations allowing latent path reconstruction:

- $\pi(x_t \mid y_{1:t-1})$ - prediction. The distribution of x_t conditioned by the previous observations $y_{1:t-1}$ is obtained by integrating over all the possible x_{t-1} values,

$$\pi(x_t \mid y_{1:t-1}) \propto \int \underbrace{\pi(x_{t-1} \mid y_{1:t-1})}_{filtering\ t-1} \cdot k(x_t \mid x_{t-1})\, d\, x_{t-1}.$$

- $\pi(x_t \mid y_{1:t})$ - filtering. The distribution of x_t conditioned by the observations up to the current position $y_{1:t}$ is given by the prediction density updated with the likelihood of current observed data y_t,

$$\pi(x_t \mid y_{1:t}) \propto \underbrace{\pi(x_t \mid y_{1:t-1})}_{prediction\ t} \cdot e(y_t \mid x_t). \tag{2.2}$$

- $\pi(x_t \mid y_{1:T})$ - smoothing. The distribution of x_t accounting for the complete sequence of observations $y_{1:T}$ may be written in terms of filtering and prediction;

$$\pi(x_t \mid y_{1:T}) \propto \underbrace{\pi(x_t \mid y_{1:t})}_{filtering\ t} \cdot \int \frac{\overbrace{\pi(x_{t+1} \mid y_{1:T})}^{smoothing\ t+1} k(x_{t+1} \mid x_t)}{\underbrace{\pi(x_{t+1} \mid y_{1:t})}_{prediction\ t+1}} d\, x_{t+1}. \tag{2.3}$$

We can write similar recursions for complete or partial trajectories $x_{1:t}$. Notably, the prediction computation writes $\pi(x_{1:t} \mid y_{1:t-1}) \propto \underbrace{\pi(x_{1:t-1} \mid y_{1:t-1})}_{filtering\ t-1} \cdot k(x_t \mid x_{t-1})$. Aiming at $x_{1:t}$ instead of x_t makes obsolete the integration over $\pi(x_{1:t-1})$.

The same type of recurrence relation may be used for the likelihood computation. The likelihood of the complete series of observations decomposes $\pi(y_{1:T}) = \pi(y_1) \prod_{t=2:T} \pi(y_t \mid y_{1:t-1})$ where each step writes as

$$\pi(y_t \mid y_{1:t-1}) = \int\int e(y_t \mid x_t) \cdot k(x_t \mid x_{t-1}) \cdot \underbrace{\pi(x_{t-1} \mid y_{1:t-1})}_{filtering\ t-1} d\, x_{t-1}\, dx_t. \tag{2.4}$$

2.1.3.1 Hidden Trajectory Reconstruction in Discrete SSMs

Non-approximate and non-asymptotic solutions for filtering, prediction, smoothing and likelihood computation tasks exist in two cases: 1) discrete state spaces and 2) linear/Gaussian observation emission and hidden kernel. We will present first a Forward-Backward algorithm for the discrete state space context and then the Kalman filter for the Gaussian context.

The classical Forward-Backward algorithm [Rabiner, 1989] computes $\pi(x_t \mid y_{1:T})$ by using forward and backward quantities

$$\pi(x_t \mid y_{1:T}) = \frac{\overbrace{\pi(x_t, y_{1:t})}^{forward\ t} \cdot \overbrace{\pi(y_{t+1:T} \mid x_t)}^{backward\ t}}{\pi(y_{1:T})}$$

Computation of forward and backward quantities involve the recurrence relations

– forward - $\pi(x_t, y_{1:t}) = \sum_{x_{t-1}} \underbrace{\pi(x_{t-1}, y_{1:t-1})}_{forward\ t-1} \cdot \pi(x_t \mid x_{t-1}) \cdot \pi(y_t \mid x_t)$,

– backward - $\pi(y_{t+1:T} \mid x_t) = \sum_{x_{t+1}} \underbrace{\pi(y_{t+2:T} \mid x_{t+1})}_{backward\ t+1} \cdot \pi(x_{t+1} \mid x_t) \cdot \pi(y_t \mid x_t)$.

Both Forward and Backward steps build on the same principles as in equation 2.2 with a sum replacing the integration. In the particle filtering literature this algorithm, with small variations, is known also as the Two Filter algorithm [Doucet and Johansen, 2009].

Algorithm 1 Forward Filtering - Backward Smoothing

Objective: Compute $\pi(x_t \mid y_{1:T})$ for each $t = 1 : T$ and state value $x_t \in \Omega_x$ (Ω_x discrete).

Forward filtering, computation of $\pi(x_t \mid y_{1:t})$:

1. For $t = 1$
   $\pi(x_1 \mid y_1) \propto \pi(x_1) \cdot e(y_1 \mid x_1)$ with $\sum_{x_1 \in \Omega_x} \pi(x_1 \mid y_1) = 1$.
2. From $t = 2$ to $t = T$
   - prediction: $\pi(x_t \mid y_{1:t-1}) \propto \sum_{x_{t-1} \in \Omega_x} \pi(x_{t-1} \mid y_{1:t-1}) \cdot k(x_t \mid x_{t-1})$; $\sum_{x_t \in \Omega_x} \pi(x_t \mid y_{1:t-1}) = 1$
   - filtering: $\pi(x_t \mid y_{1:t}) \propto \pi(x_t \mid y_{1:t-1}) \cdot e(y_t \mid x_t)$; $\sum_{x_t \in \Omega_x} \pi(x_t \mid y_{1:t}) = 1$.

Backward smoothing, computation of $\pi(x_t \mid y_{1:T})$:

1. For $t = T$ use $\pi(x_T \mid y_{1:T})$ computed during forward filtering.
2. From $t = T - 1$ to $t = 1$ compute
   $\pi(x_t \mid y_{1:T}) = \pi(x_t \mid y_{1:t}) \cdot \sum_{x_{t+1} \in \Omega_x} \frac{\pi(x_{t+1} \mid y_{1:T}) \cdot k(x_{t+1} \mid x_t)}{\pi(x_{t+1} \mid y_{1:t})}$.

The algorithm 1 is a variant of the Forward Backward algorithm that uses filtering and smoothing recurrence relations (2.2 and 2.3) and illustrates for a discrete state space our approach for the continuous case.

2.1.3.2 Hidden Trajectory Reconstruction in Continuous Gaussian SSMs

In the continuous space, exact solutions exist for Gaussian SSMs. The Kalman Filter [Kalman, 1960] deals in its standard description with such models governed both in latent space and in observation measurement by Gaussian distributions. In a simple form the SSM writes

$$\begin{aligned} x_t &= ax_{t-1} + v_t \quad \text{i.e.} \quad x_t \sim \mathcal{N}(ax_{t-1}, \sigma_v^2) \\ y_t &= hx_t + w_t \quad \text{i.e.} \quad y_t \sim \mathcal{N}(hx_t, \sigma_w^2), \end{aligned}$$

where y_t and x_t are the observations and latent variables and v_t and w_t are white measurement errors of standard deviations σ_v and σ_w.

The Kalman filter (KF) provides a reconstruction of $x_t \mid y_{1:t}$ that minimizes the mean square error (MSE). It involves two steps at each position of the sequence:

1. Prediction $x_t^* = a\hat{x}_{t-1}$.

2. Correction

$$\hat{x}_t = x_t^* + \underbrace{(y_t - y_t^*)}_{\text{measurement residual}} \cdot \underbrace{\frac{\text{var}(x_t^*)h}{\text{var}(y_t^*)}}_{\text{Kalman Gain}} . \tag{2.5}$$

Here, x_t^* represents the prediction based on $\hat{x}_{t-1}$ and the latent chain transition, $\hat{x}_t$ the KF estimate obtained after the correction of x_t^* based on the observation y_t and $y_t^* = hx_t^*$ the predicted observation. The variance $\text{var}(y_t^*)$ may be computed as $\text{var}(\text{E}(y_t^* \mid x_t^*)) + \text{E}(\text{var}(y_t^* \mid x_t^*)) = \sigma_v^2 h^2 + \sigma_w^2$ according to the law of total variance. The variance of x_t^* is σ_v^2. Intuitively we first make predictions based only on the transition rule and then we correct them with the observed gap from the data. We weight this gap by the proportion of variance in data prediction explained by the variance in the hidden chain prediction.

Within a Bayesian framework the path reconstructed with the KF corresponds also to a MAP estimate (discussed by [Chen, 2003]). For a Gaussian SSM we can derive analytically the marginal distributions associated with prediction, filtering and smoothing problems. To illustrate this, we write the update step $\pi(x_t \mid y_t, x_{t-1}) \propto \pi(y_t \mid x_t) \cdot \pi(x_t \mid x_{t-1})$ which

coincides with the KF recursion we presented above

$$\begin{aligned}\pi(x_t \mid y_t, x_{t-1}) &\propto \exp\left(-\frac{(x - a\,x_{t-1})^2}{2\sigma_v^2}\right) \cdot \exp\left(-\frac{(y - h\,x_t)^2}{2\sigma_w^2}\right) \\ &\propto \exp\left(-\frac{\sigma_w^2 + \sigma_v^2 h^2}{2\sigma_v^2\sigma_w^2}\left(x - \frac{a\,x_{t-1}\sigma_w^2 + y\,h\,\sigma_v^2}{\sigma_w^2 + \sigma_v^2 h^2}\right)^2\right) \\ &= \mathcal{N}\,(x_t; mean = \frac{a\,x_{t-1}\sigma_w^2 + y\,h\,\sigma_v^2}{\sigma_w^2 + \sigma_v^2 h^2}, sd^2 = \frac{\sigma_v^2\sigma_w^2}{\sigma_w^2 + \sigma_v^2 h^2}\,)\end{aligned}$$

where we denote $\mathcal{N}(x; \mu, \sigma^2)$ the density at x of $\mathcal{N}(\mu, \sigma^2)$. The mode of the posterior density and the minimum MSE are obtained for $\hat{x}_t = \frac{a\,x_{t-1}\sigma_w^2 + y_t\,h\,\sigma_v^2}{\sigma_w^2 + \sigma_v^2 h^2} = a\,x_{t-1} + (y_t - a\,x_{t-1}\,h)\frac{h\sigma_v^2}{\sigma_w^2 + \sigma_v^2 h^2}$ and is the same as the KF estimator from 2.5.

Extended Kalman Filter deals with more complex models by using Taylor approximations. Switching linear models, that have a natural adaptation to change point detection, can be approached using the Kalman filter principles [Zymnis et al., 2008].

2.2 Sequential Monte Carlo

The Sequential Monte Carlo (SMC) methods are a new class of Monte Carlo based solutions to sequential problems (in the general case the approximation of $\{\pi(x_{1:t})\}_{t\geq 1}$) that are particularly useful for to SSM inference. Further on we will use in this manuscript SMC with reference only to SSM models, i.e. for the target density $\pi(x_{1:T} \mid y_{1:T})$. The principles of Monte Carlo methods are based on the laws of large numbers. The approximation $\pi_{MC}(x_{1:T} \mid y_{1:T})$ of $\pi(x_{1:T} \mid y_{1:T})$ is obtained as

$$\pi_{MC}(x_{1:T} \mid y_{1:T}) = \sum_{p=1:P} \delta_{\{x_{1:T}^p\}}(x_{1:T}),$$

where $(x_{1:T}^p)_{1:P}$ is a sample of size P drawn (asymptotically) from the target $\pi(x_{1:T} \mid y_{1:T})$.

As sampling directly a R^T (R is the real space) state space becomes impossible even for small values of T, the trajectories $x_{1:T}^p$ are build in a sequential manner. In practice the SMC methods for SSMs make use of the recurrence relations described in section 2.1.3. If update quantities raise sampling issues, as it is usually the case, the Importance sampling (IS) techniques can provide effective solutions. In an IS scheme, the target density $\pi(x_{1:T} \mid y_{1:T})$ is approximated by using a weighed sample $(w^p, x_{1:T}^p)_{1:P}$, drawn from an instrumental function (named also proposal or importance function) $q(x_{1:T})$. The IS weights are computed as $w^p = \frac{\pi(x_{1:T}^p|y_{1:T})}{q(x_{1:T}^p)}$. The IS principles are grounded on on a simple relation $\int \pi(x_{1:T} \mid y_{1:T})\, dx_{1:T} = \int \frac{\pi(x_{1:T}|y_{1:T})}{q(x_{1:T})}\, dq(x_{1:T})$ which permits the approximation

$$\pi_{IS}(x_{1:T} \mid y_{1:T}) = \sum_{p=1:P} w^p \delta_{\{x_{1:T}^p\}}(x_{1:T}).$$

2.2.1 Sequential Importance Resampling - SIR

In a typical SMC filtering algorithm, the Sequential Importance Resampling (SIR) [Gordon et al., 1993; Doucet and Johansen, 2009] the sample of P particles has a sequential, Monte Carlo build-up. In the algorithm 2, at each position t=1:T the approximation of $\pi(x_{1:t} \mid y_{1:t})$ builds on a sample of P particles $(w^p, x_{1:t}^p)_{p=1:P}$. These particles are constructed up to position $t-1$ from the SMC approximation of $\pi(x_{1:t-1} \mid y_{1:t-1})$ and extended to position t in an IS manner. This construction associates to each particle the IS weight w_t^p. The particle-weight association $(x_{1:t}^p, w_t^p)_{p=1:P}$ represents (asymptotically) a weighted sample from $\pi(x_{1:t} \mid y_{1:t})$ and approximates it by $\pi_{SMC}(x_{1:t}) =$

$\sum_{p=1:P} w_t^p \delta_{\{x_{1:t}^p\}}(x_{1:t})$. Resampling according to the weights $(w_t^p)_{p=1:P}$ leads to an unweighted approximation $\sum_{p=1:P} \delta_{\{x_{1:t}^{a_t^p}\}}(x_{1:t})$. Here we introduce an additional variable a_t^p defined in the particle index space $1 : P$ and that accounts for the resampled indices and thus the particle ancestry. Resampling is not required for the algorithm validity and alternative algorithms like Sequential Importance Sampling (SIS) exist where weights are simply propagated along each of the particles.

Algorithm 2 Sequential Importance Resampling

Objective: recursively approximate $x_{1:t} \mid y_{1:t}$ for t=1:T.

1. For $t = 1$ and for $p = 1 : P$:
   (a) draw $x_t^p \sim q_1(x_1)$ where $q_1(x_1)$ is a proposal density,
   (b) compute the unnormalized $w_1^p = \frac{\pi(y_1|x_1^p)\pi(x_1^p)}{q_1(x_1^p)}$ and normalized $W_1^p = \frac{w_1^p}{\sum_{p=1:P} w_1^p}$ weights.
2. For t=2:T and for p=1:P,
   (a) draw the index of the ancestor particle a_{t-1}^p from the weights $\left(W_{t-1}^p\right)_{1:P}$ such as $P(a_{t-1}^p = k) = W_t^k$,
   (b) draw $x_t^p \sim q_t(x_t; x_{t-1}^{a_{t-1}^p})$,
   (c) compute the unnormalized particle weights $w_t^p = \frac{\pi(x_t^p|x_{t-1}^{a_{t-1}^p})\pi(y_t|x_t^p)}{q_t(x_t^p;x_{t-1}^{a_{t-1}^p})}$ and normalized weights $W_t^p = \frac{w_t^p}{\sum_{p=1:P} w_t^p}$.

The P particles ending in $(x_T^p)_{1:P}$ represent asymptotically a weighted sample from $\pi(x_T \mid y_{1:T})$ with weights $(W_T^p)_{1:P}$. Furthermore, a complete trajectory $\tilde{x}_{1:T}^p$ for a particle p can be reconstructed by backtracking.

3. Set $p_T = p$ the index at position T of the particle to be reconstructed and assign $\tilde{x}_T^p = x_T^{p_T}$. For $t = T$ to $t = 2$ set $p_{t-1} = a_t^{p_t}$ and $\tilde{x}_{t-1}^p = x_{t-1}^{p_{t-1}}$.

The trajectories $(\tilde{x}_{1:T}^p, W_T^p)_{p=1:P}$ represent a weighted sample of $\pi(x_{1:T} \mid y_{1:T})$.

The proposal can be a position-specific density conditioned on the local observations and previous sampled values: $q_t(x_t; x_{t-1}) = f(x_t; x_{t-1}, y)$. The alternative algorithm SIS does

not include the resampling step 2a. With this change, the particle weights need to account for all the past update weights. For SIS the tasks in step 2 therefore change to

(a') draw $x_t^p \sim q_t(x_t; x_{t-1}^p)$,

(b') compute $w_t^p = W_{t-1}^p \cdot \frac{\pi(x_t^p|x_{t-1}^p)\pi(y_t|x_t^p)}{q_t(x_t^p;x_{t-1}^p)}$ and $W_t^p = \frac{w_t^p}{\sum_{p=1:P} w_t^p}$.

The SIS reconstruction based on steps a' and b' provides a weighted sample approximation

$$\pi_{SMC}(x_{1:t} \mid y_{1:t}) = \sum_{1:P} W_t^p \cdot \delta_{\{x_{1:t}^p\}}(x_{1:t}). \tag{2.6}$$

The reconstruction using importance sampling and thus instrumental densities can possibly lead to particles with very low importance weights (W_t^p). Propagating these particles has two detrimental effects: computation resources are consumed on unlikely trajectories and the size of the final particle sample representativeness will be reduced. This motivates the resampling step 2a which can be considered as a filter to discard unfitted particles. After resampling the approximation can be done using the unweighed sample $\left(x_{1:t}^{a_t^p}\right)_{1:P}$ and it writes

$$\pi_{SMC}(x_{1:t} \mid y_{1:t}) = \frac{1}{N}\sum_{1:P} \delta_{\{x_{1:t}^{a_t^p}\}}(x_{1:t}). \tag{2.7}$$

2.2.1.1 The Resampling Step: How and When

The drawing of $(a_t^p)_{p=1:P}$ indices needs to preserve the particle weights: $P(a_t^p = k) = W_t^k$. A first option consists in independent draws of indices and sums to a multinomial sampling scheme. After drawing the number of particle off-springs $n^{1:P}$ by multinomial sampling $n^{1:P} \sim \mathcal{M}(P, W_t^{1:P})$, indices can be assigned in order to $(a_t^p)_{p=1:P}$ as $a_t^{\sum_{j=1:p-1} n^j : \sum_{j=1:p} n^j} = p$. This option, even if valid, has a significant additional variance (computed as excess of the 2.7 over the 2.6 estimators). This motivated methods that tend to generate for each particle a number of off-springs n^p close to the expected number $W_t^p \cdot P$. The systematic and residual sampling implement this idea. While both methods ensure a minimum number of off-springs for the particle p equal to $\lfloor W_t^p \cdot P \rfloor$ (the integer part of $W_t^p \cdot P$), the procedures differ.

Systematic sampling consists in drawing an initial value $u \sim \mathcal{U}(0,1)$ and setting the number of off-springs n_t^p to the cardinal value of the set $\{ \ell : P \cdot \sum_{j=1:p-1} W_t^j < u + \ell \leq$

$P \cdot \sum_{j=1:p} W_t^j\}$. We can first compute $n_t^1 = \lfloor P \cdot W_t^1 + 1 - u \rfloor$ and then for $p = 2 : P$ compute recursively $n_t^p = \lfloor P \cdot W_t^p - (\sum_{j=1:p-1} n_t^j - 1 + u - \sum_{j=1:p-1} P \cdot W_t^p) \rfloor$.

In Residual sampling first set for each particle an initial number of off-springs $n'^p_t = \lfloor W_t^p \cdot P \rfloor$. In a second step draw the residual $R = P - \sum_{j=1:P} \lfloor W_t^j \cdot P \rfloor$ off-springs (n''^p_t for each particle). The $n''^{1:P}_t$ can be drawn using a multinomial (or systematic) sampling scheme with weights $\frac{W_t^p - \lfloor W_t^p \cdot P \rfloor}{R}$. The final number of off-springs is $n_t^p = n'^p_t + n''^p_t$.

More details on these resampling techniques are discussed in [Hol et al., 2006; Chopin and Singh, 2013].

While the standard SIR involves resampling at each position, its frequency can be modulated without affecting the validity of the algorithm. In SMC literature ([Doucet and Johansen, 2009] among others), a frequent criteria pertains to sample representativeness. The Effective Sampling Size value $ESS = 1/\sum_{p=1:P} (W^p)^2$ (described in [Liu, 2008]) measures the changes in the variance by using the proposal instead of the target and in other words the loss in efficiency.

To explain in more detail the ESS we look at the estimation efficiency for the expectation of a generic function $h(x)$ ($E = \int h(x)\, d\pi(x)$). We consider two estimators $\hat{E}_{\pi_{ESS}}$ and $\hat{E}_{\hat{\pi}}$ where the first one is obtained using an unweighted sample of size ESS drawn under $\pi(x)$ and the later a weighted sample $(W, x)^{1:P}$, of size P, drawn under an instrumental distribution q. The ESS gives the size of a sample drawn under $\pi(x)$ for which the variance of the two estimators is approximatively equal, i.e. $\int (\hat{E}_{\pi_{ESS}} - E)^2\, d\pi(x) \approx \int (\hat{E}_{\hat{\pi}} - E)^2\, dq(x)$. In a more simple interpretation $\sum_{p=1:P} (W^p)^2$ is the probability of drawing the same sample index in two independent drawings for a population with weights $W^{1:P}$. This probability is $\frac{1}{P}$ for an unweighted sample of size P. Thus, the ESS also represents the size of an unweighted sample with a probability $\sum_{p=1:P} (W^p)^2$ of drawing consecutively the same index.

The resampling frequency choice is a matter of algorithm efficiency and has no influence on its asymptotic accuracy and other criteria may be considered (like the number of particles with weights under a threshold or the maximum weight value).

2.2.1.2 Backward Sampling

The resampling step induces a degeneracy phenomenon: the coalescence of trajectories associated to the particles. After a number of resampling events all the present particles have the same ancestor, i.e. the hidden trajectories that can be backtracked for each of them coalesce.

To get a hint of the time before coalescence we look at a simplified SSM with no observations and for which sampling is done under the transition kernel ($q(x_t^p; x_{t-1}^p) = \pi(x_t^p \mid x_{t-1}^p)$). In this case, at each step, the weights $w_t^p = \frac{\pi(x_t^p|x_{t-1}^p)}{q(x_t^p;x_{t-1}^p)}$ equal to 1, and, for a multinomial resampling procedure, the average Time to the Most Recent Common Ancestor (TMCRA) of all the particles is less or equal to $2P \cdot (1 - \frac{1}{P})$ [Mohle, 2004]. Because more uneven weights lead to faster coalescence, in practice proposals have a significant impact on the time to coalescence.

Forward Filtering provides a particle sample $(x_{1:T}^p)_{1:P}$ from $\pi(x_{1:T} \mid y_{1:T})$ but, due to coalescence the approximation in one SIR run would use only one particle for positions before $T - TMCRA$. Even if asymptotically correct, the filtering can thus not provide a good approximation with practical P and for high T values.

We can reconsider in a backward algorithm the values proposed during filtering and thus suppress the coalescence effects. The Backward Smoothing [Godsill et al., 2004; Doucet and Johansen, 2009] builds upon equation 2.3 which, for a filtering approximation $\pi(x_t \mid y_{1:t}) \approx \sum_{p=1:P} W_t^p \delta_{\{x_t^p\}}(x_t)$ leads to the smoothed approximation $\pi_b(x_t \mid y_{1:T}) = \sum_{p=1:P} W_t^p \delta_{\{x_t^p\}}(x_t) \cdot \int \frac{\pi(x_{t+1}|y_{1:T}) k(x_{t+1}|x_t^p)}{\pi(x_{t+1}|y_{1:t})} \, dx_{t+1}$. From this we can derive the smoothed weights $W_{t|T}^p$ of the x_t^p sampled values,

$$W_{t|T}^p = W_t^p \cdot \sum_{l=1:P} \frac{W_{t+1|T}^l \cdot k(x^l{}_{t+1} \mid x_t^p)}{\sum_{k=1}^P W_t^k \cdot k(x^l{}_{t+1} \mid x_t^k)},$$

that allow the Backward Smoothing approximation $\pi_b(x_t \mid y_{1:T}) \approx \sum_{p=1:P} W_{t|T}^p \, \delta_{\{x_t^p\}}(x_t)$.

A related option is the Backward Sampling (algorithm 3) [Godsill et al., 2004; Doucet and Johansen, 2009; Chopin and Singh, 2013] which backward reconstructs the trajectories reconsidering the filtering samples. It starts from the decomposition $\pi(x_{1:T} \mid y_{1:T}) = \pi(x_T \mid y_{1:T}) \prod_{t=T-1:1} \pi(x_t \mid x_{t+1}, y_{1:T})$ and recursively extends backward the $x_{t:T}^*$ trajectory according to the $\pi(x_t \mid x_{t+1}, y_{1:T}) \propto \pi(x_t \mid y_{1:t}) \cdot k(x_{t+1} \mid x_t)$ approximation

$$\pi_b(x_t \mid x_{t+1}^*, y_{1:T}) \propto W_t^p \cdot k(x_{t+1}^* \mid x_t^p).$$

Similarly we can write the weights $W^p_{t|t+1} \propto W^p_t \cdot k(x^*_{t+1} \mid x^p_t)$, where $\sum_{p=1:P} W^p_{t|t+1} = 1$, and the approximation $\pi_b(x_t \mid x^*_{t+1}, y_{1:T}) \approx \sum_{p=1:P} W^p_{t|t+1}\, \delta_{x^p_t}(x_t)$.

The approximation of $\pi(x_t \mid y_{1:T})$ is computed directly during Backward Smoothing and as a marginal from $\pi(x_{1:T} \mid y_{1:T})$ in Backward Sampling.

Algorithm 3 Backward sampling

Objective: backward sample a trajectory $x^*_{1:T}$ from the sample $(x^p_{1:T}, w^p_{1:T})_{1:P}$ generated with the SIR particle filter.

For $t = T$,

1. Draw the index $p^\star_T$ from $1:P$ indices with $(w^p_T)_{1:P}$ weights.
   Set $x^*_T = x^{p^*_T}_T$.

For $t = T-1:1$,

1. For $p = 1:P$ compute $w^p_{t|t+1} \propto w^p_t \cdot \pi(x^p_t \mid x^*_{t+1})$ weights of $\pi(x^p_t \mid x^*_{t+1}, y_{1:T})$
2. Draw the index $p^\star_t$ from $1:P$ indices with weights $\left(w^p_{t|t+1}\right)_{1:P}$. Set $x^*_t = x^{p^*_t}_t$

We show empirical results that illustrate the Backward sampling improvement in results, section 4.2.4.

2.2.2 Particle Markov Chain Monte Carlo - PMCMC

Even with backward smoothing, the SMC approximation is not precise for a practical number of particles and thus highly influenced by the instrumental function. To tackle with the exactness issue recent methods use the convergence properties of the MCMC algorithms. The Particle MCMC (PMCMC) algorithms provide exact sampling methods for finite sample sizes P and for any valid instrumental function q by plugging the SMC approximation $\pi_{SMC}(x_{1:T} \mid y_{1:T})$ as a proposal in a MCMC algorithm. These iterative algorithms allow to obtain a correlated sample $\left(x_{1:T}^{(n)}\right)_{n\geq 1}$ distributed, after a period of burn in, under the target $\pi(x_{1:T} \mid y_{1:T})$. Further on in this chapter we simplify, in order to ease reading, the writing for the complete sequences $x_{1:T}$, $y_{1:T}$ to x and y.

The Metropolis Hastings (MH), a MCMC algorithm on which the PMCMC is built, aims to construct a Markov chain $(x^n)_{n\geq 1}$ with the target $\pi(x)$ as stationary distribution. The transition $k(x^{(n)}; x^{(n-1)})$ is built in two steps:

1. draw $\tilde{x}$ from an instrumental function q: $\tilde{x} \sim q(x; x^{(n-1)})$,
2. set $x^{(n)} = \tilde{x}$ with probability $\rho(\tilde{x}; x^{(n-1)}) = \min\left(1, \frac{q(x^{(n-1)};\tilde{x})\cdot\pi(\tilde{x})}{q(\tilde{x};x^{(n-1)})\cdot\pi(x^{(n-1)})}\right)$ and $x^{(n)} = x^{(n-1)}$ otherwise.

The Markov chain transition kernel is $k(x^{(n)}; x^{(n-1)}) = q(x^{(n)}; x^{(n-1)}) \cdot \rho(x^{(n)}; x^{(n-1)}) + \delta_{\{x^{(n-1)}\}}(x^{(n)}) \int q(x^{(n)}; x^{(n-1)}) \cdot (1 - \rho(x^{(n)}; x^{(n-1)}))\, dx^{(n)}$. It satisfies the detailed balance $\pi(x^{(n-1)}) \cdot k(x^{(n)}; x^{(n-1)}) = \pi(x^{(n)}) \cdot k(x^{(n-1)}; x^{(n)})$ and therefore it has $\pi(x)$ as stationary density.

A MH algorithm for SSM with a SMC particle approximation as instrumental function would write as:

1. propose $x_{1:T}^{*} \sim \pi_{SMC}(x_{1:T} \mid y_{1:T})$,
2. set $x_{1:T}^{(n)} = x_{1:T}^{*}$ with probability $\min\left(1, \frac{\pi_{SMC}(x_{1:T}^{(n-1)}|y_{1:T}) \cdot \pi(x_{1:T}^{*}|y_{1:T})}{\pi_{SMC}(x_{1:T}^{*}|y_{1:T}) \cdot \pi(x_{1:T}^{(n-1)}|y_{1:T})}\right)$ and otherwise set $x_{1:T}^{(n)} = x_{1:T}^{(n-1)}$.

In practice though, we do not have access neither to $\pi_{SMC}(x_{1:T}^{(n-1)} \mid y_{1:T})$ and $\pi_{SMC}(x_{1:T}^{*} \mid y_{1:T})$ nor directly to $\frac{\pi_{SMC}(x_{1:T}^{(n-1)}|y_{1:T})\cdot\pi(x_{1:T}^{*}|y_{1:T})}{\pi_{SMC}(x_{1:T}^{*}|y_{1:T})\cdot\pi(x_{1:T}^{(n-1)}|y_{1:T})}$ and therefore we cannot compute the acceptance probability and thus we cannot build the Markov chain. While we can compare the particles reconstructed within the same SMC run (through the particle weights), $x_{1:T}^{(n-1)}$ and $x_{1:T}^{*}$ are obtained in different runs.

2.2.3 Particle Independent Metropolis Hastings - PIMH

The acceptance probability can be replaced by $\min\left(1, \frac{\pi^*_{SMC}(y_{1:T})}{\pi^{(n-1)}_{SMC}(y_{1:T})}\right)$, where the $\pi^{(n)}_{SMC}(y_{1:T})$ is a likelihood approximation from the nth SMC run [Andrieu et al., 2010].

In the SMC framework and with the current notations the likelihood writes: $\pi^{(n)}_{SMC}(y_{1:T}) = \prod_{t=1:T} \frac{1}{P} \sum_{p=1:P} w_t^{p(n)}$ where $(w_t^{p(n)})_{p=1:P}$ are the unnormalized particles weights at position t computed during the nth SMC run. This formula can be derived from equation 2.4 that, after a change in the integration density to $q(x)$ and using the filtering approximation of $\pi(x_{t-1} \mid y_{1:t-1})$ by an unweighted sample $x_{t-1}^{1:P}$, writes as $\pi(y_t \mid y_{1:t-1}) \approx \frac{1}{P} \sum_{p=1:P} \frac{e(y_t|x_t^p) \cdot k(x_t^p|x_{t-1}^p)}{q(x^p; x_{t-1}^p)}$ and consequently $\pi_{SMC}(y_t \mid y_{1:t-1}) = \frac{1}{P} \sum_{p=1:P} w_t^p$

In [Andrieu et al., 2010], the authors establish that the PIMH update create a Markov chain converging to $\pi(x_{1:T} \mid y_{1:T})$. They obtain this result by considering this density as the marginal in a state space augmented by all SMC variables.

Algorithm 4 Particle Independent Metropolis Hastings

Objective: build a $\left(x^{(n)}_{1:T}\right)_{n \geq 1}$ correlated sample drawn from $\pi(x_{1:T} \mid y_{1:T})$.

For iteration $n = 1 : N$

1. Run SMC (SIR) targeting $\pi(x_{1:T} \mid y_{1:T})$
   (a) compute $\pi_{SMC}(y_{1:T}) = \prod_{t=1:T} \frac{1}{P} \sum_{p=1:P} w_t^p$,
   (b) sample a particle $x^*_{1:T}$ according to the particle weights $(W_T)^{p=1:P}$.
2. With probability $\min\left(1, \frac{\pi_{SMC}(y_{1:T})}{\pi^{(n-1)}_{SMC}(y_{1:T})}\right)$ set $x^{(n)}_{1:T} = x^*_{1:T}$ and $\pi^{(n)}_{SMC}(y_{1:T}) = \pi_{SMC}(y_{1:T})$. Otherwise, set $x^{(n)}_{1:T} = x^{(n-1)}_{1:T}$ and $\pi^{(n)}_{SMC}(y_{1:T}) = \pi^{(n-1)}_{SMC}(y_{1:T})$.

Intuitively, we build the Markov chain $x^{(n)}_{n \geq 1}$ by always accepting trajectories drawn from SMC run associated with higher estimated values of the likelihood and reject with some probability those with a smaller likelihood estimated values.

2.2.4 Conditional SMC Update - CSMC

Another way to deal with the comparison of two particles $x_{1:T}^{(n-1)}$ and $x_{1:T}^{*}$ is to have them both in the same SMC run. In this case the comparison can be done using the importance weights.

Algorithm 5 Conditional SMC update

Objective: build a correlated sample $\left(x_{1:T}^{(n)}\right)_{n\geq 1}$ such that $\pi(x_{1:T}^{(n)} \mid y_{1:T})$ is preserved. The Markov chain is built on a kernel $K_{CSMC}(x_{1:T}^{(n)}; x_{1:T}^{(n-1)}, y_{1:T})$ that draws $x_{1:T}^{(n)}$ from a SMC that has the trajectory $x_{1:T}^{(n-1)}$ as one of the particles.

1. For $t = 1$
   (a) For $p = 1 : P-1$ draw x_1^p from the proposal density $q_1(x_1^p)$.
   Set $x_1^P = x_1^{(n-1)}$.
   (b) For $p = 1 : P$ compute the weights: unnormalized $w_1^p = \frac{\pi(y_1|x_1^p)\pi(x_1^p)}{q(x_1^p)}$ and normalized $W_1^p = \frac{w_1^p}{\sum_{p=1:P} w_1^p}$.
2. For $t = 2$ to $t = T$
   (a) For $p = 1 : P-1$ draw index of the ancestor particle a_{t-1}^p from weights $\left(W_{t-1}^p\right)_{1:P}$ such as $P(a_{t-1}^p = k) = W_{t-1}^k$. Set the ancestor of the last particle $a_{t-1}^P = P$.
   (b) For $p = 1 : P-1$, draw x_t^p from the proposal $q_t(x_t^p; x_{t-1}^{a_{t-1}^p})$. Set $x_t^P = x_t^{(n-1)}$.
   (c) For $p = 1 : P$, compute the weights: $w_t^p = \frac{\pi(y_t|x_t^p)\cdot\pi(x_t^p|x_{t-1}^{a_{t-1}^p})}{q_t(x_t^p; x_{t-1}^{a_{t-1}^p})}$ and $W_t^p = \frac{w_t^p}{\sum_{p=1:P} w_t^p}$.
3. Backtrack $x_{1:T}^{(n)}$ from $\left(x_{1:T}^p, w_{1:T}^p, a_{1:T-1}^p\right)_{1:P}$:
   (a) Draw p_T from $1 : P$ indices with weights $(w_T^p)_{1:P}$ and set $x_T^{(n)} = x_{t-1}^{p_T}$.
   (b) For $t = T$ to $t = 2$ do: $p_{t-1} = a_{t-1}^{p_t}$ and $x_{t-1}^{(n)} = x_{t-1}^{p_{t-1}}$.

Setting one particle to the trajectory from the $n-1$ sweep might result, when sampling the final particles, in the same trajectory for the sweep n. Moreover, even if we sampled a different particle, this new trajectory will be the same as the old trajectory for positions

before the last common particle ancestor (see coalescence in section 2.2.1.2).

In the place of the backtrack step (algorithm 5, step 3) we can use the backward sampling (algorithm 3) that reconstructs a trajectory $x_{1:T}^{(n)}$ from $(x_{1:T}^p, w_{1:T}^p)_{1:P}$ and thus suppresses this behaviour that impedes the MCMC mixing a long sequence. A second option to tackle this issue is to make the update by blocks, i.e. each update is done on segments of reduced size T. We present in the results chapter implementations of the block update and backward sampling CSMC.

2.3 Parameter Estimation for SSMs

Until this point we focused on the reconstruction of the hidden trajectory $x_{1:T}$ from the observations $y_{1:T}$ at supposed known values of parameters Θ (emission and transition parameters). In a practical case, the parameters are often not known and we need to estimate them.

The direct analysis of the complete data likelihood $\pi(y_{1:T} \mid \Theta)$ and posterior $\pi(\Theta \mid y_{1:T})$ is generally not possible because it involves a high dimensional integration $\pi(y_{1:T} \mid \Theta) = \int \pi(y_{1:T}, x_{1:T} \mid \Theta)\, dx_{1:T}$. In most cases, the same quantities for $\pi(y_{1:T}, x_{1:T} \mid \Theta)$ and $\pi(\Theta \mid x_{1:T}, y_{1:T})$ are much more tractable. In both the Bayesian and Maximum Likelihood frameworks, the algorithms for the estimation of parameters (based on $\pi(y_{1:T} \mid \Theta)$ and $\pi(\Theta \mid y_{1:T})$) are intimately connected to the reconstruction of the hidden trajectory.

2.3.1 Maximum Likelihood Estimators

Within the Maximum Likelihood framework, we present Stochastic Expectation Maximisation (SEM) [Celeux et al., 1995] and Maximum Likelihood via Iterator Filtering (MIF) [Ionides et al., 2006] algorithms. SEM simplifies the computation required by the Expectation Maximisation algorithm by using Monte Carlo approximations during the expectation step. MIF implements a different option, that involves parameter update during trajectory reconstruction, and results in local parameter optimisations. The local estimations are used to derive global parameter values.

Maximum Likelihood (ML) methods aim to maximise the likelihood probability of observed data $\Theta_{ML} = \underset{\theta}{\operatorname{argmax}}\, \pi(y \mid \theta)$. Maximum A Posteriori (MAP)methods are an extended version of ML with a prior on the parameters: $\Theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\, \pi(y \mid \Theta) \cdot \pi(\Theta)$. The Expectation Maximisation is a classic iterative algorithm for models with latent variables that aims to increase the likelihood at each sweep. The algorithm decomposes in two steps and writes for a SSM as:

1. E-step: compute $Q(\Theta; \Theta^{(n)}) = \int \log(\pi(x_{1:T}, y_{1:T} \mid \Theta)) \cdot \pi(x_{1:T} \mid y_{1:T}, \Theta^{(n)})\, dx_{1:T}$
2. M-step: compute $\Theta^{(n+1)} = \arg\max_{\Theta} Q(\Theta; \Theta^{(n)})$

As we already mentioned, the complete data likelihood $\pi(x_{1:T}, y_{1:T} \mid \Theta)$ is easier to compute than the incomplete data likelihood $\pi(y_{1:T} \mid \Theta)$.

The models where the E-Step requires tedious or intractable integration to compute $Q(\Theta;\Theta^{(n)})$, this quantity can be approached with Monte Carlo approximations. The Stochastic EM (SEM) and Monte Carlo EM (MCEM), described among several other stochastic EM methods in Celeux et al. [1995], reduce the difficulty of computing $Q(\Theta;\Theta^{(n)})$ to sampling $\pi(x_{1:T} \mid y_{1:T}, \Theta)$. In MCEM sample of size P is used to approximate computation of the expectation. High values of P lead to estimations closer to deterministic EM. For $P = 1$ the MCEM reduces to the SEM algorithm. The SMC algorithms are a viable alternative to generate such samples [Lindsten, 2013]. In a SEM algorithm, for a SMC approximation, the E-step integration of $Q(\Theta;\Theta^{n})$ is replaced by the simulation of one trajectory. This simplifies the M-step to maximizing the complete data likelihood $\pi(x_{1:T}^{(n)}, y_{1:T} \mid \Theta)$. Due to its stochastic E-step, the SEM algorithm builds a Markov chain $\left(\Theta^{(n)}\right)_{n>1}$ that, instead of converging to a point value like the EM does, converges under mild assumptions to a stationary distribution around the ML value [Nielsen, 2000].

Algorithm 6 Stochastic EM

1. Set a random initial value for $\Theta^{(1)}$;
2. Repeat (until convergence):
   (a) Stochastic E-step: sample $x_{1:T}^{(n)}$ from $\pi(x_{1:T} \mid y_{1:T}, \Theta^{(n)})$
   (b) M-step: compute $\Theta^{(n+1)} = \underset{\Theta}{\operatorname{argmax}}\, \pi(\Theta \mid x_{1:T}^{(n)}, y_{1:T})$

The Maximum Likelihood via Iterator Filtering (MIF) [Ionides et al., 2006; Breto et al., 2009] uses another strategy to build a $\Theta^{(n)}$ series converging to the ML. A few precisions need to be made concerning the MIF algorithm that we present for illustration purpose (algorithm 7). First, in the original algorithm [Breto et al., 2009] the initial value x_0 is estimated and included as a separate variable. Beside simplification we omit x_0 also because such a state does not fit our model. In algorithm 7, we sample the x_1 values from the distribution $\pi(x_1 \mid \Theta)$. Second, as in the original algorithm we present here sampling done by a prior kernel proposal. We believe that using a more complex proposal would not invalidate its properties if the importance weights are modified correspondingly. Third,

the parameter Θ can be multivariate (we omitted this for simplicity purposes). In this case σ becomes a covariance matrix Σ and steps 2b, 2c and 3 are done for each component of Θ.

Algorithm 7 Maximum Likelihood via Iterator Filtering

For $n = 1$ initialize randomly $\Theta^{(1)}$;
For n in $2 : N$

1. For $t = 1$
   (a) Set $\bar{\Theta}_0 = \Theta^{(n)}$ draw $\Theta_0^p \sim \mathcal{N}(\Theta^{(n)}, b\sigma^{(n)})$;
2. For t in $1 : T$
   (a) For p in $1 : P$ sample
   - draw $x_t^p \sim \pi(x_t^p \mid x_{t-1}^{a_{t-1}^p}, \Theta_{t-1}^p)$; If $t = 1$ draw from $x_t^p \sim \pi(x_t^p \mid \Theta_{t-1}^p)$;
   - compute $W_t^p \propto e(y_t \mid x_t^p, \Theta_{t-1}^p)$ such as $\sum_{p=1:P} W_t^p = 1$;
   - draw a_t^p particle index from weights $(w_t^p)_{p=1:P}$
   - draw $\Theta_t^p \sim \mathcal{N}(\Theta_{t-1}^{a_t^p}, \sigma^n)$;

   (b) compute $\bar{\Theta}_t = \frac{1}{N} \sum_{p=1:P} \Theta_{t-1}^{a_t^p}$ the sample mean of $\left(\Theta_{t-1}^{a_t^p}\right)_{p=1:P}$
   (c) compute $V_t = \frac{1}{N} \sum_{p=1:P} (\Theta_t^p)^2 - (\frac{1}{N} \sum_{p=1:P} \Theta_t^p)^2$ the variance mean of $(\Theta_t^p)_{p=1:P}$
3. $\Theta^{(n+1)} = \Theta^{(n)} + V_1 \cdot \sum_{t=1:T} \frac{1}{V_t} (\bar{\Theta}_t - \bar{\Theta}_{t-1})$

The maximum likelihood estimate for Θ is obtained as $\Theta^{(N)}$. Here, the super(sub)-scripts (n), p and t refer to : the sweep n, the particle p and the position t. a_t^p represents the index of the p particle after resampling; $\sigma^{(n)}$ represents the parameter proposal variance $\sigma^{(n)} = s^{n-1}\sigma$ with $0 < s < 1$, and $b > 0$ represents an initial scaling factor for the same variance.

2.3.2 Parameter Estimation with PMCMC Methods

Within the Bayesian framework we present two recent methods: Particle Marginal Metropolis Hastings (PMMH) and Particle Gibbs [Andrieu et al., 2010].

2.3.2.1 Particle Marginal Metropolis Hastings

If we adopt a Monte Carlo Markov chain strategy that aims at sampling the joint distribution $x, \Theta \mid y$ and makes a joint update at the sweep n, the standard Metropolis Hastings acceptance ratio for the proposed values (x^*, Θ^*) drawn from the proposal $q(x, \Theta; x^{(n-1)}, \Theta^{(n-1)})$ writes

$$\min\left(1\,,\ \frac{q\left(x^{(n-1)}, \Theta^{(n-1)}; x^*, \Theta^*\right)\ \cdot\ \pi(x^*, \Theta^* \mid y)}{q\left(x^*, \Theta^*; x^{(n-1)}, \Theta^{(n-1)}\right)\ \cdot\ \pi(x^{(n-1)}, \Theta^{(n-1)} \mid y)}\right). \tag{2.8}$$

The joint density $\pi(x, \Theta \mid y)$ decomposes as $\pi(x \mid y, \Theta) \cdot \pi(\Theta \mid y)$. Our quantity of interest is $\Theta \mid y$ and we assume that $x \mid \Theta, y$ is well aproximated by the SMC runs. This suggests using the proposal $q(x^*, \Theta^*; x^{(n-1)}, \Theta^{(n-1)}) = \pi_{_{SMC}}(x^* \mid y, \Theta^*) \cdot q_\Theta(\Theta^*; y, \Theta^{(n-1)})$. The acceptance ratio (equation 2.8) would write $\frac{\pi_{SMC}(x^{(n-1)}|y,\Theta^{(n-1)})\cdot q_\Theta(\Theta^{(n-1)};y,\Theta^*)\cdot\pi(x^*|y,\Theta^*)\cdot\pi(\Theta^*|y)}{\pi_{SMC}(x^*|y,\Theta^*)\cdot q_\Theta(\Theta^*;y,\Theta^{(n-1)})\cdot\pi(x^{(n-1)}|y,\Theta^{(n-1)})\cdot\pi(\Theta^{(n-1)}|y)}$ and for a perfect SMC approximation, i.e. $\pi_{_{SMC}}(x \mid y, \Theta) = \pi(x \mid y, \Theta)$ would simplify to

$$\frac{q_\Theta(\Theta^{(n-1)}; y, \Theta^*) \cdot \pi(\Theta^* \mid y)}{q_\Theta(\Theta^*; y, \Theta^{(n-1)}) \cdot \pi(\Theta^{(n-1)} \mid y)} \tag{2.9}$$

We write $\pi(\Theta \mid y_{1:T}) \propto \pi(y_{1:T} \mid \Theta) \cdot \pi(\Theta)$. The left term of the product is the likelihood we compute in equation 2.4 and approximate it in section 2.2.3 for a SIR algorithm as $\pi_{_{SMC}}(y_{1:T} \mid \Theta) = \prod_{t=1:T} \frac{1}{P} \sum_{p=1:P} w_t^p$, ($w^p$ unnormalized weights).

In [Andrieu et al., 2010], the authors prove that the Particle Marginal Metropolis Hasting (algorithm 8), a Metropolis Hastings - algorithm with the acceptance ratio from the equation 2.9 computed using the likelihood SMC approximation indeed leaves $\pi(\Theta \mid y_{1:T})$ invariant despite the fact that the SMC approximation is not perfect.

Algorithm 8 Particle Marginal Metropolis Hastings

Objective: build a correlated sample $\left(\Theta^{(n)}\right)_{n=1:N}$ from the target $\pi(\Theta \mid y_{1:T})$.

For iteration $n = 1 : N$

1. Draw Θ^* from $q_\Theta(\Theta^*; \Theta^{(n-1)})$.
2. Run SMC for $x \mid y, \Theta^*$ to compute $\pi_{SMC}(y \mid \Theta^*) = \prod_{t=1:T} \frac{1}{P} \sum_{p=1:P} w_t^p$.
3. With probability $\min\left(1, \frac{q_\Theta(\Theta^{(n-1)};\Theta^*) \cdot \pi_{SMC}(y|\Theta^*) \cdot \pi(\Theta^*)}{q_\Theta(\Theta^*;\Theta^{(n-1)}) \cdot \pi_{SMC}^{(n-1)}(y|\Theta^{(n-1)}) \cdot \pi(\Theta^{(n-1)})}\right)$ set $\Theta^{(n)} = \Theta^*$ and $\pi_{SMC}^{(n)}(y \mid \Theta^{(n)}) = \pi_{SMC}(y \mid \Theta^*)$
   If not, set $\Theta^{(n)} = \Theta^{(n-1)}$ and $\pi_{SMC}^{(n)}(y \mid \Theta^{(n)}) = \pi_{SMC}^{(n-1)}(y \mid \Theta^{(n-1)})$

PMMH efficiency (mixing in $\left(\Theta^{(n)}\right)_{n\geq 1}$) highly depends on the SMC likelihood approximation (see also P. Fearnhead discussion on [Andrieu et al., 2010]).

2.3.2.2 Particle Gibbs

In a general Gibbs algorithm, each sweep updates the joint $\pi(x_{1:T}, \Theta \mid y_{1:T})$ in two steps:

1. update $x_{1:T} \mid \Theta$ and $y_{1:T}$,
2. update $\Theta \mid x_{1:T}$ and $y_{1:T}$.

In Particle Gibbs the trajectory update is done through the CSMC algorithm. The parameter update can recurs to various sampling techniques preserving $\Theta \mid x_{1:T}, y_{1:T}$. Such techniques include sampling from the conditional posterior distribution (directly, sometimes referred to as Gibbs-type move or mediated by importance densities) and MCMC methods (like MH for example). For multivariate Θ we can adopt a Gibbs strategy and proceed to the update of parameters by blocks instead of a joint update of all the parameters.

Algorithm 9 Particle Gibbs

Objective: build a sample $\left(x_{1:T}^{(n)}, \Theta^{(n)}\right)_{n\geq 1}$ from the target distribution $\pi(\Theta, x_{1:T} \mid y_{1:T})$.

For iteration $n = 1 : N$

1. Run CSMC targeting $\pi(x_{1:T} \mid y_{1:T}, \Theta^{(n-1)})$ and draw one trajectory $x_{1:T}^{(n)}$
2. Update $\Theta^{(n)}$ such as $\pi(\Theta \mid x_{1:T}^{(n)}, y_{1:T})$ is preserved.

2.4 The Choice of the Instrumental Function in Importance Sampling

The fitness of the instrumental function q (also referred to as the proposal or importance function) to a target distribution π is a crucial factor in any Monte Carlo estimation. In SMC notably, the Importance Sampling schemes make use intensively of the proposal functions (named in this context importance functions). The general Importance Sampling approximation uses the equivalence of $\pi(x)$ with $\frac{\pi(x)}{q(x)} \cdot q(x)$. The quantities $w(x) = \frac{\pi(x)}{q(x)}$ represent the importance weights (for which we use w^p notation for the weight of x^p sampled value). The importance distribution q can be virtually any function that dominates the target π distribution (i.e. $q(x) > 0$ for $\pi(x) > 0$). To ensure finite variance $\int (\frac{h(x)f(x)}{q(x)})^2 \, dq(x) - (\int \frac{h(x)f(x)}{q(x)} \, dq(x))^2$ we usually choose $q(x)$ such as the ratio $\frac{h(x)f(x)}{q(x)}$ is bounded, i.e. $\exists M \in \mathbb{R}$ such as $\frac{h(x)f(x)}{q(x)} < M$ [Cappe et al., 2005; Doucet and Johansen, 2009]. The choice of a proposal may be done with a regard to computational efficiency, fitness to target function and variance of estimators.

Computational efficiency is influenced by the sample size and the costs of proposal sampling and density computation. In the Sequential Importance Sampling filtering framework for SSM the general target is $\pi(x_t \mid x_{t-1}, y_t) \propto \pi(x_t \mid x_{t-1}) \cdot \pi(y_t \mid x_t)$. Its product writing suggests proposals that address each term. A classic choice is the prior kernel [Cappe et al., 2005] $q(x_t; x_{t-1}) = \pi(x_t \mid x_{t-1})$. The computation of the weights simplifies in this case to $w_t = \pi(y_t \mid x_t)$.

It is important to notice that the choice of the prior kernel eliminates the need of density computation for the hidden kernel. This decreases the cost of computation. For SSMs where these densities are not available, the partial black-box SSMs, these proposal choices might be the only practical solutions. In a general scenario though, as they include only partial information, they often require high sample sizes for target approximation.

In classical IS, the fitness of a proposal is measured by a "distance", like the relative entropy (Kullback-Leibler divergence) or derived from the Effective Sample Size (discussed in section 2.2.1), between target and proposal. The Kullback-Leibler (KL) writes $\mathcal{KL}(\pi||q) = \int \ln \frac{\pi(x)}{q(x)} \pi(x) \, dx$. Its approximation by a sample $(x^p)_{p=1:P}$ generated under $q(x)$ writes $\mathcal{KL}(\pi||q) \approx \sum_{p=1:P} \ln \frac{\pi(x^p)}{q(x^p)} \cdot w^p$.

To improve proposal fitness a proposal parameter tuning step might be added in SMC

algorithms. We consider a general mixture proposal $q(x;\alpha,\Theta) = \sum_{d=1:D} \alpha_d \cdot q_d(x)$ where in this section $\alpha = (\alpha_d)_{d=1:D}$ and $\Theta = (\Theta_d)_{d=1:D}$. Two classes of parameters can be optimized: the mixture parameters α and proposal component parameters Θ. We will discuss an adaptive method, the Population Monte Carlo (PMC) implemented in versions aiming at optimizing the mixture weights (D-kernel PMC [Douc et al., 2007]) or both the mixture weights and the parameters (MPMC) [Cappe et al., 2007]. In an adaptive algorithm the proposal function is updated according to the fitness of previous obtained samples to the target. The principle of PMC resides on the fact that updating the proposal does not jeopardize the fundamental importance sampling identity.

Algorithm 10 Mixture Population Monte Carlo

Objective: estimate the parameters $(\alpha = (\alpha_d)_{d=1:D}$ and $\Theta = (\Theta_d)_{d=1:D}$ of a mixture proposal $q(x;\alpha,\Theta) = \sum_{d=1:D} \alpha_d\, q_d(x;\Theta_d)$ that maximize $\int \ln q(x;\alpha,\Theta)\pi(x)\,dx$.

For iteration n , $n = 1:N$,

1. Generate a sample $(x^p)_{p=1:P}$ from the $n-1$ proposal mixture $q(x;\alpha^{(n-1)},\Theta^{(n-1)})$
2. Compute the importance weights $w^p = \frac{\pi(x^p)}{q(x^p;\alpha^{(n-1)},\Theta^{(n-1)})}$ and $W^p = \frac{w^p}{\sum_{p=1:P} w^p}$
3. Compute the mixture posterior probabilities

$$\rho(x^p;\alpha_d^{(n-1)},\Theta_d^{(n-1)}) = \frac{\alpha_d^{(n-1)} \cdot q_d(x^p;\Theta_d^{(n-1)})}{q(x^p;\alpha^{(n-1)},\Theta^{(n-1)})}$$

4. Update the weights $\alpha_q^{(n)} = (\alpha_1^{(n)}..\alpha_D^{(n)})$:

$$\alpha_d^{(n)} = \sum_{p=1:P} W^p \cdot \rho(x^p;\alpha_d^{(n-1)},\Theta_d^{(n-1)})$$

5. Update the parameters $\Theta^{(n)} = (\Theta_1^{(n)}..\Theta_D^{(n)})$:

$$\Theta_d^{(n)} = \underset{\Theta_d}{\operatorname{argmax}} \sum_{p=1:P} W^p \rho(x^p;\alpha_d^{(n-1)},\Theta_d^{(n-1)})\, \ln q_d(x^p;\Theta_d)$$

The parameters α^N and Θ^N maximize $\int \ln q(x;\alpha,\Theta)\pi(x)\,dx$ and minimize the KL divergence of $q(x;\alpha,\Theta)$ from $\pi(x)$, $\int \ln \frac{\pi(x)}{q(x;\alpha,\Theta)}\pi(x)\,dx = \int \ln \pi(x)\,d\pi(x) - \int q(x;\alpha,\Theta)\pi(x)\,dx$.

Chapter 3

A New Model for RNA-Seq Read Counts

We begin the results section by exploring the diversity of the data issued from RNA-Seq protocols. For this analysis we include datasets from some of the most studied species of bacteria and small eukaryotes. In the most simple scenario, where variability would be introduced only by random sampling of the molecules that are sequenced, reads should have an uniform distribution over regions with the same expression level and read counts should follow a Poisson distribution. The experimental data-sets we study indicate that the read counts distribution is more complex than that. We show results on nine data-sets concerning two types of biases: one related related to random local sequence scaling and the other due to position within a transcript. We pay a particular attention to the possibility of a bias associated to the RNA conformation and its relation to the presence of G and C nucleotides. We observe that bias is reproducible for data sets with the same origin but can manifest in different forms for different organisms and studies. To circumvent difficulties that could be induced by explicitly including all bias sources, we rather design a robust over-dispersed read count model.

In our first attempts we try to account for the variability of counts inside regions with expected homogeneous coverage by using a Negative Binomial (NB) distribution. However, we observe a significant discrepancy between the data and the NB with respect to the relation between mean, variance and the percentage of positions that have no counts (zero-counts) within a transcript. This prompted us to search for a more accurate model

that would make sense from a mechanistic perspective. Within the canonical RNA-Seq flow we consider the randomness introduced at each step and build an intricate mixture model for the count distribution.

3.0.1 RNA-Seq Datasets

Data-set	SRA	Publication	Protocol*	Use
B. anthracis	SRR028684	Passalacqua et al. [2009]	s/36**	♮
B. subtilis	SRR064325	Lasa et al. [2011]	s/36	♮
C. albicans 1	SRR492978	Vandeputte et al. [2012]	s/79	♮♮
C. albicans 2	NA	NA	s/50	♮♮
C. albicans 3	NA	NA	s/50	♮♮
E. coli 1	SRR915698	Li et al. [2013]	s/100	♮♮
E. coli 2	SRR794856	McClure et al. [2013]	s/100	♮♮♮
E. coli 3	SRR794832	McClure et al. [2013]	s/100	♮♮
F. psychrophilum 1	NA	NA	s/50	♮
F. psychrophilum 2	NA	NA	s/50	♮
H. phylori	SRR031126	Sharma et al. [2010]	s/76	♮♮
P. acnes	SRR850805	Lin et al. [2013]	s/75***	♮♮
S. aureus	SRR397557	Lasa et al. [2011]	p/76	♮♮
S. cerevisiae 1	SRR121907	Dijk et al. [2011]	s/50****	♮♮♮∇
S. cerevisiae 2	SRR1042851	Guydosh and Green [2014]	s/50	♮♮
S. cerevisiae 3	SRR815616	Rouskin et al. [2014]	s/50	♮♮
S. cerevisiae 4	SRR927162	Dijk et al. [2011]	s/50****	∇

Table 3.1: List of data-sets analysed in this thesis. * the default protocol is Illumina ;** no strand specificity ;*** FRT-Seq data. **** SOLiD data. Except *S. aureus* data set (paired-end) sequencing was done on single reads of length indicated in the protocol column. The "use" column indicates the type of analysis performed on the data-sets: ♮ we estimated parameters of the new read count model , ♮♮ in addition we analysed read count distribution bias, ♮♮♮ in addition we evaluated the approximation of transcription borders using Parseq, and ∇ these data-sets were used in evaluating differential expression results. *C. albicans* 2-3 and *F. psychrophilum* 1- 2 are unpublished Illumina datasets that were provided respectively by the Laboratory of Computational and Quantitative Biology, Paris (F. Devaux group) and by the Mathematics, Informatics and Genome Laboratory, Jouy-en-Josas, France.

3.0.2 Notations

Throughout the following chapter 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);
- $\mathcal{NB}(m,\kappa)$ is the Negative Binomial distribution with mean m and overdispersion κ;
- $\Gamma(m,\kappa)$ is the Gamma distribution with mean m and shape $\frac{1}{\kappa}$ (scale= $m\cdot\kappa$, var= $m^2\cdot\kappa$);
- $\mathcal{U}(a,b)$ is the uniform distribution between a and b;
- $\mathbf{1}_{\{c\}}$ and $\mathbf{1}_{\{c\}}(x)$ are the indicator functions whose values are 1 if condition c is true, and respectively if x=c, and 0 otherwise;
- δ_d is the Dirac delta function with unit mass at point d.

3.1 Bias in RNA-Seq Read Counts

The RNA-Seq protocol consists in several steps that generate randomness and might introduce bias in the read distribution within a transcript. Bias factors may be intrinsic to transcripts or pertaining to the protocol itself. Intrinsic bias might be related to the transcript's sequence and to other transcript characteristics (length, secondary structure).

We investigate in this section the bias on 9 datasets (*E. coli* 1-3 , *C. albicans* 1-3 and *S. cerevisiae* 1-3). We show briefly results related to the influence of the sequence composition in regions surrounding the 5' end of a read and to the position relative to transcript boundaries. We continue with a study on the impact of RNA conformation (the secondary structure).

The nucleotides composing the sequences surrounding the 5'-end of a read were shown to correlate to specific scaling of read counts (section 1.4). We computed coefficients for each nucleotide type in a window w of lag 10 on each side by using the Poisson regression option from `mseq` [Li et al., 2010]. `Mseq` uses a Poisson model of mean x_{gt} parametrized as $\ln x_{gt} = \ln x_g + \sum_{k \in w} \sum_{n \in \{A,C,G\}} \xi_n^k \mathbf{1}_{\{n\}}(n_{t+k})$ where g represents the index of homogeneous expression units, t is a genomic position within g, n_t represents the nucleotide at the position t, $\mathbf{1}_{\{n\}}(n_{t+k})$ equals 1 if $n_{t+k} = n$ and 0 otherwise and ξ_n^k is the scaling coefficient for nucleotide n at position k relative to the read 5'-end. The Mseq coefficients ξ have base level for the nucleotide T, $\xi_T^k = 0$. Here, these coefficients were centred such that, taking into account the chromosome composition, the average value $e^{\xi_n^k}$ is 1. We show in figure 3.1 scaled coefficients: $\tilde{\xi}_k^n = \xi_k^n - \ln(\sum_n e^{\xi_k^n})$.

The results presented in (figure 3.1) suggest that there is a high influence of positions in a window of ± 5 bp. This influence is highly correlated for datasets issued from the same series of experiments (≈0.98 correlation of $\xi_T^{-10:10}$ for *C. albicans* 2 vs. 3 and *E. coli* 2 vs. 3) but not necessarily correlated among datasets from the same organism.
Initial versions of our application to estimate transcript boundaries, Parseq, incorporated the scaling coefficients computed from the $\tilde{\xi}_k^n$'s in the emission function (section 4.1.1). We computed for each position a scaling coefficient ξ_t as the sum of the coefficients for all the nucleotides in the window $\xi_t = \sum_{k \in w} \sum_{\mathrm{n} \in \{\mathrm{A,C,G,T}\}} \tilde{\xi}_k^n \mathbf{1}_{\{n\}}(n_{t+k})$. We included these into our emission function by simply scaling the emission expectation by ξ_t.

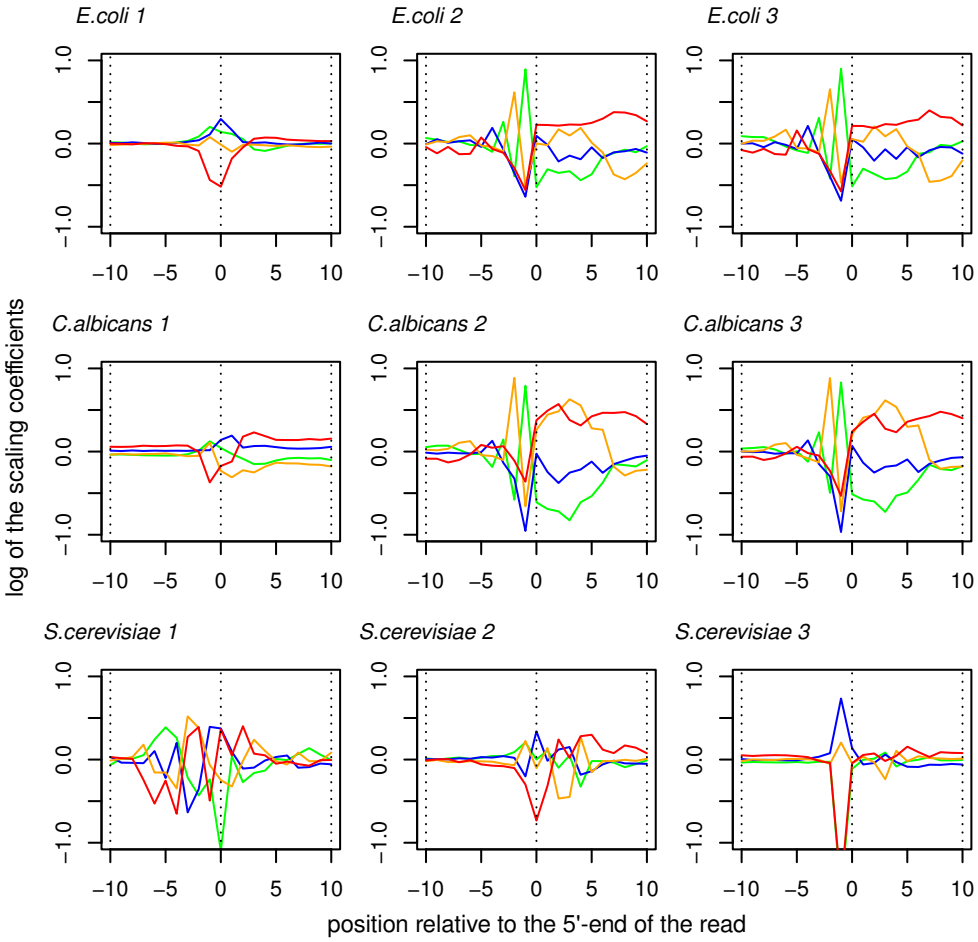

Figure 3.1: Local sequence coefficients computed with a Poisson linear model (`mseq`) for a window of 21 bp around the 5' end of the read. Colours: green - A, blue - C, orange - G, red - T. Y-axis: logarithms of scaling coefficients ξ (ex.: for *E. coli* 2 a nt. T (red) at position -1 scales the expected counts by $e^{\tilde{\xi}_k^T} = e^{0.95}$).

The **longitudinal bias** is also widely commented in the literature (section 1.4). We investigated read coverage close to ORF boundaries (figure 3.2). The change in coverage at the 5'-ends and 3'-ends ranges from very small (*S. cerevisiae* 1) to significant progression upward (*S. cerevisiae* 3) or significant sudden changes (*C. albicans* 1). Our main work relates to estimating transcript boundaries based on differences in read coverage. We show in section 4.1.2 that including a component (denoted drift) to account for progressive changes improves the results. Significant sudden changes, as we notice in *C. albicans* 1 or *E. coli* 2 may lead to the incorrect identification of transcript boundaries and are partially addressed by a local correlation variable (denoted s in section 3.2). Importantly, the central transcript regions seem to have an uniform coverage. We use these regions when evaluating emission model parameters (section 3.3).

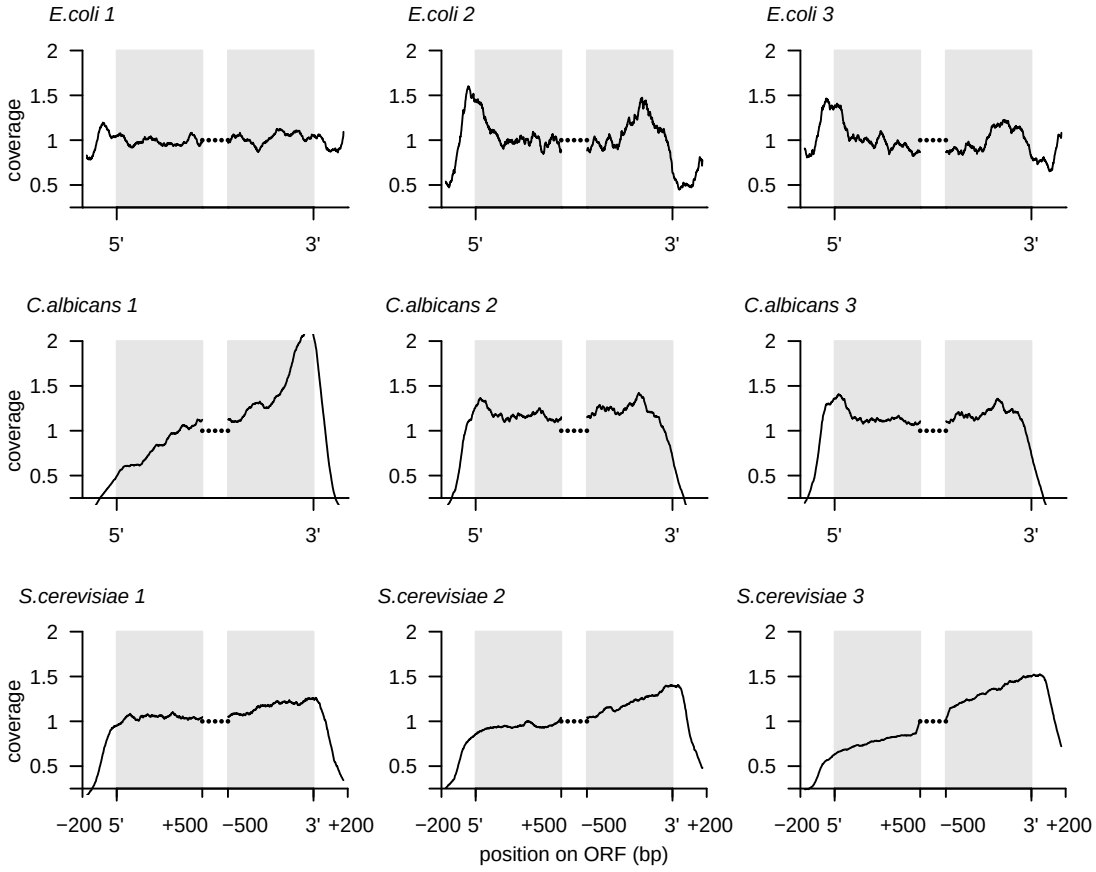

Figure 3.2: Count coverage on the first and last 500 bp of ORFs (grey areas) and on 200 bp upstream/downstream the ORF borders. For each position in an ORF we computed the average in a 51 bp window centred on the position and then scaled this value by the ORF average. Then, we average over the ORFs with lengths in 1000 - 2000 bp range and coverage above 0.5 reads/bp.

3.1.1 Bias Induced by RNA Conformation and GC Content

We are interested in evaluating the possible influence of the RNA conformation on RNA-Seq data. We speculate that, due to its high local correlation, the RNA secondary structures might be at the origin of the local correlations observed in several datasets (figure 3.7). We focus on investigating the RNA secondary structure impact on read counts and specifically we ask the question if transcript positions that are part of a stems, i.e. involved in a base-pair, have different count expectation. We define a variable ρ_t for the preference of a position to be in a base-pair, $\rho_t \in [0, 1]$ with $\rho_t = 1$ when the position t is in a base-pair in all conditions and $\rho_t = 0$ if the position t is always in a loop.

A transcript might adopt several conformations and these are often unstable. Also, few genome-wide experimental determinations of the RNA secondary structures are currently available. We start this analysis by relying on predicted transcript conformation. We use `RNA2Dfold`, a minimum free energy prediction application from the ViennaRNA package [Lorenz et al., 2011] to approximate secondary structures in ORF regions. Transcript length and temperature play determining roles in transcript conformation. We do not have access to these informations and design a method that tries to obtain robust secondary structure estimations independent of temperature and transcript length. For this we predict structures for different transcript lengths (ORFs and ORFs extended on each side with 50, 100 and 150 bp) and 4 values for the temperature parameter: 30°C, 35°C, 37°C and 40°C. We derive the values of the ρ_t by averaging the 16 predictions.

We do a second analysis using experimental genome-wide pairing scores for *S. cerevisiae* obtained with Parallel Analysis of RNA Structure (PARS) [Kertesz et al., 2010]. Genome-wide pairing scores were derived from a differential analysis of two RNA-Seq count profiles obtained using two different enzymes: RNase V1 which preferentially cleaves phosphodiester bonds from the 3'-ends of double-stranded RNA and RNase S1 which preferentially cleaves phosphodiester bonds from the 3'-ends of single-stranded. We use pair scores computed in [Kertesz et al., 2010] (SRP003175) from merged score profiles.

We perform the analysis of count (y_t) distribution for given pairing preferences (ρ_t) for 6 data sets (*E. coli* 1-2 , *C. albicans* 1-2 and *S. cerevisiae* 2-3). PARS data were included in the analysis for *S. cerevisiae* 2-3. For all datasets we consider only ORFs with lengths between 500 and 2000 bp and having a mean 5'-end read count per position above 0.5

and at least 200 positions with positive counts. For a rough analysis of the base-paring conditional count expectation, $E(y_t \mid \rho_t)$, we compute estimates for three ranges of ρ_t values: $[0, 0.1]$, $(0.1, 0.9]$, $(0.9, 1]$.

Results, illustrated for *S. cerevisiae* 2 in figure 3.3 (top left), suggest that there is a significant difference between the expectation of counts for positions with high base-pairing preference ($\rho_t > 0.9$) and those with low preference ($\rho_t < 0.1$). Similarly, on results not included in the figure 3.3, we observe that the count expectation is with 20% lower for high base-pair preference positions for *E. coli* 1 , *C. albicans* 1 and *S. cerevisiae* 3 data-sets. Surprisingly, we notice a significant difference but in the opposite sense (25% higher count expectation in high base-pair preference positions) for *C. albicans* 2. We did not register any difference for *E. coli* 2. Importantly, this difference does not depend on the method used for the approximation of the base-pair profile $\rho_{1:T}$ and is similar for the RNA conformation predicted to minimize the free energy (RNAfold) and for the experimental data (PARS).

Next, we try to investigate the mechanisms underlying the correlation of the RNA conformation with the counts. The GC base-pair has stronger bounds than AT or AU base-pairs with three hydrogen bonds instead of two. Therefore the G and C positions (GC further-on) might have a higher probability to be in a stem structure (in our data $\frac{E_{GC_t=1}(\rho_t)}{E_{GC_t=0}(\rho_t)} \approx 1.1$ and $\frac{E_{\rho_t>0.9}(GC_t)}{E_{\rho_t<0.1}(GC_t)} \approx 1.3$ for both RNAfold and PARS data). Also, GC rich regions are known to have potentially biased read counts (see section 1.4). These suggest that the variable GC_t (indicating the presence of G or C at position t) could explain partially read count bias associated to the base-pairing.

First, we confirm that the counts have a significant difference between GC and AT positions. For *E. coli* 1 , *S. cerevisiae* 2-3 and *C. albicans* 1 the GC positions have in average up to 30% less counts (illustrated only for *S. cerevisiae* 2 in figure 1.4). Interestingly, for *E. coli* 2 and *C. albicans* 2 the GC positions have up to 30% more reads and for *S. cerevisiae* 1 no bias was observed.

Second, we try to disentangle the influence of GC and ρ on read count. For this, we examine the read distribution within the three ranges of base-pair preference separately for GC and AT positions. We show results in figure1.4 for *S. cerevisiae* 2 (PARS base-pairing approximation) and *E. coli* 1-2 (RNAfold base-pairing approximation). The count expectation estimators are computed as $\hat{\mu}_{i,j} = exp\{\sum_g \ln \frac{\hat{\mu^g}_{\rho_t \in R_i, GC=j}}{\hat{\mu^g}_{\rho_t \in R_1, GC=0}}\}$, where $\hat{\mu^g}_{\rho_t \in R_i, GC=j}$ repre-

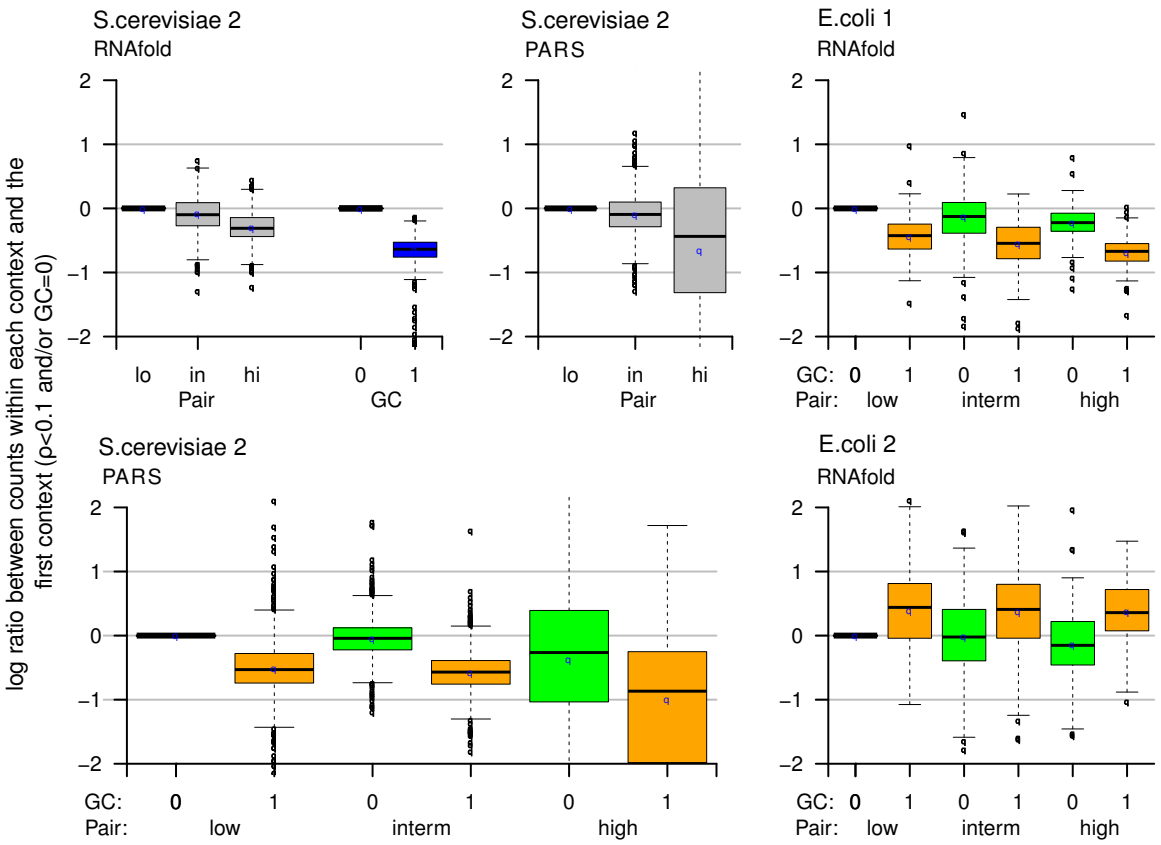

Figure 3.3: Base-pairing and GC correlation with the read counts. The box-plots summarize data at the ORF level for different contexts. Contexts distinguished for ρ_t: low($\rho_t \leq 0.1$), interm ($\rho_t \in (0.1, 0.9]$) and high ($\rho_t \in (0.9, 1]$). Contexts distinguished for the GC status: 0 if AT (green) and 1 if GC (orange).

sents the average of counts for positions with ρ_t in range $R_i \in \{[0, 0.1], (0.1, 0.9], (0.9, 1]\}$ and GC content j for the ORF g.

We observe that count averages for AT and GC positions taken separately (green and orange box plots) have significantly smaller changes with increasing base-pair preference than when the two CG contexts are pooled. Without providing sufficient quantitative proof, it suggests that the secondary structure and specifically the base-pairing has a weak influence on read counts once the GC influence is removed. The observed correlation of base-pairing with the read count bias is due mainly to the base-pairing correlation with the GC content. There is still a residual bias that either might point to an influence of the secondary structure or might be explained by analysing the base-pairing correlation to a more complete sequence composition.

3.2 A Mechanistic Model for RNA-Seq Read Counts

The variability of read counts observed when re-sequencing the same library has been described as almost compatible with a Poisson distribution [Marioni et al., 2008]. However, when compared between samples (or even replicate libraries), the distribution exhibits over-dispersion and the negative binomial (NB) distribution is often used to accommodate this behavior [Robinson et al., 2010; Anders and Huber, 2010]. RNA-Seq counts often show a high overdispersion (example in figure 3.4). Like Cleynen et al. [2014a], we relied on the NB in our first attempts of modelling the count distribution within transcripts. Indeed, it seems required to involve a mixed Poisson distribution in order to account simultaneously for the incompressible variance of the final sampling by sequencing (Poisson) and for the extra-variability introduced by randomness in library preparation and by position-specific biases that can be introduced at all steps of the protocols. In this context, the NB, viewed as a Gamma-Poisson mixture ($y_t \sim \text{Poisson}(x_t z_t)$ where z_t follows a Gamma distribution with mean 1 and variance ϕ), stands as the most tractable model [Karlis and Xekalaki, 2005].

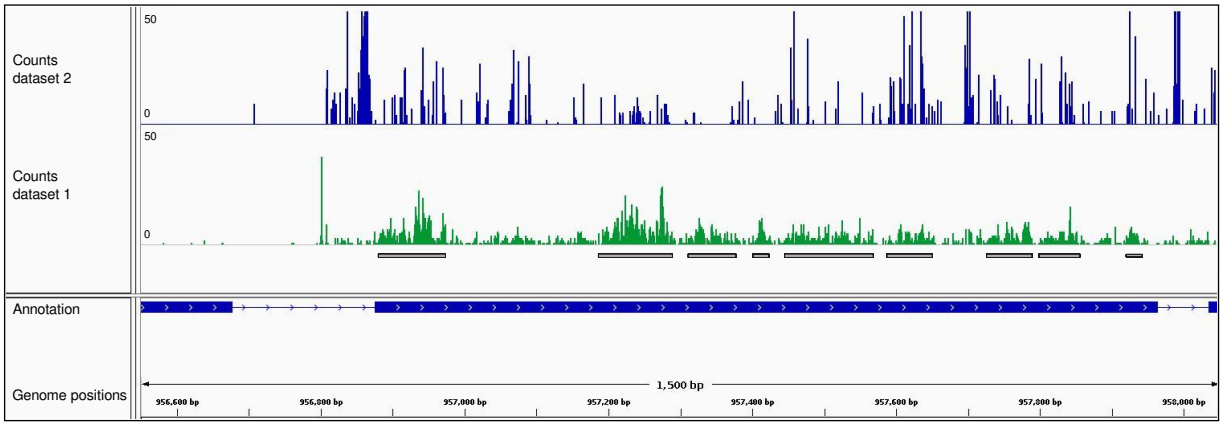

Figure 3.4: Two examples of the distribution of RNA-Seq read counts on a region of 1500 bp on the Watson strand of *E. coli* 1 (data-set 1) and *E. coli* 2 (data-set 2). Blue and green bars: counts of read 5' ends. Grey lines: segments with relatively homogeneous read coverage in the data-set 2. Blue boxes: annotated transcripts.

Based on two real data-sets, we examined the distribution of read-counts inside Open Reading Frames (ORF), regions expected to be homogeneous in terms of expression level. Namely, we asked whether the NB could capture the relationships between mean and

variance and simultaneously account for the fraction of positions with zero-counts (figure 3.5). Both characteristics are expected to impact directly on the decision to predict read-counts at distant positions as originating from the same transcript. The most obvious discrepancy between the data and the NB is with respect to the zero-counts: given the mean and the variance of the empirical distribution, the fraction of positions with zero-counts under the NB assumption tends to be too low for low expression levels and too high for high expression levels.

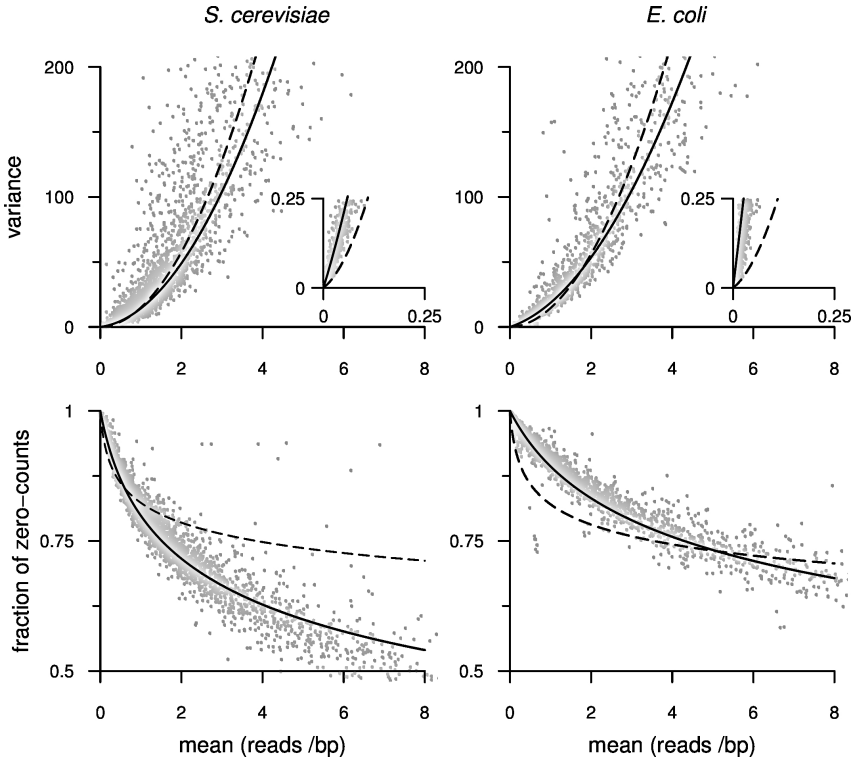

Figure 3.5: Read count variance and zero fraction within regions of homogeneous expression (ORFs). Data sets: *S. cerevisiae* 1 (left) *E. coli* 2 (right). Each long ORF (region without in-frame stop codon) identified on the genome is represented by a dot. Dashed lines show the fit of the negative binomial model with overdispersion parameter estimated via variance (reads^2/bp^2) versus mean (reads/bp) regression; plain lines show the fit with the Parseq model.

The usual parametrization of the NB with overdispersion parameter ϕ mentionned above is also contradicted by the data. Indeed, the variance increases markedly faster than the mean x even for very low expression level, in sharp contrast with the prediction that the variance should write $x + \phi x^2$. In the Poisson-mixture context, breaking this behaviour, that arises from law of total variance, implies that the relationship between

the mixing distribution and x is more subtle than a simple scaling. This prompted us to search for a more accurate model that would make sense from a mechanistic perspective.

We developed a new RNA-Seq read count emission model that fits much better the characteristics of the real data than the simple NB (figure 3.5). Its construction intends to account for the three main steps of the experimental protocol (figure 3.6): (i) initial molecule sampling and fragmentation, (ii) amplification, and (iii) final sampling by sequencing.

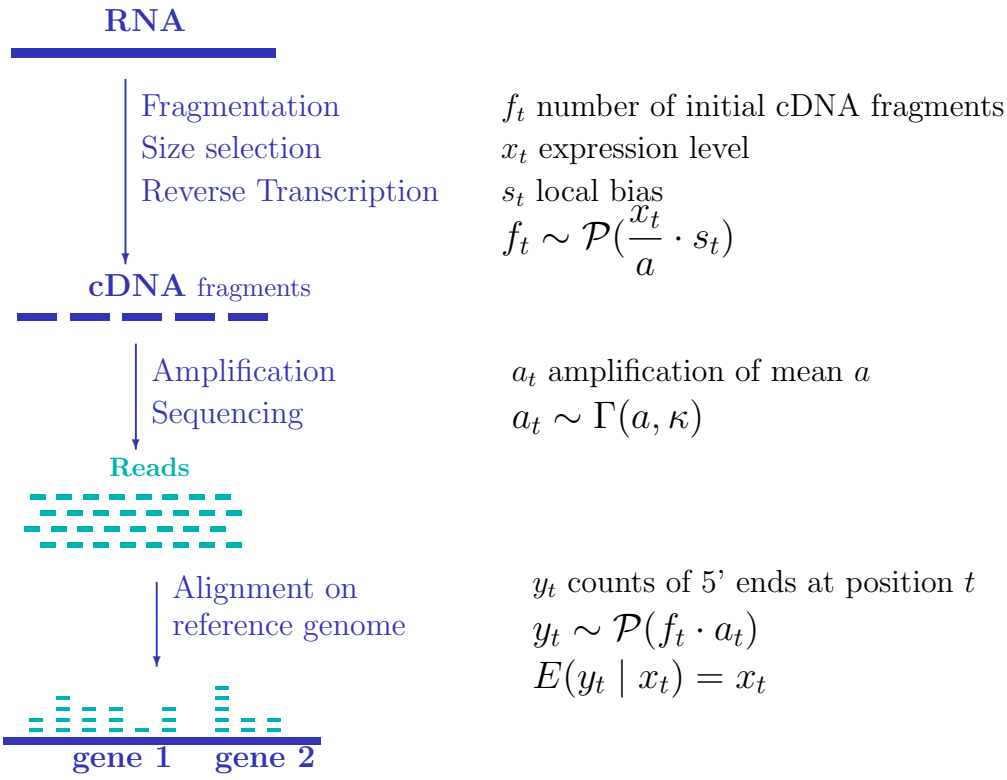

Figure 3.6: Canonical RNA-Seq protocol steps including: i) RNA fragmentation and Reverse Transcription; ii) cDNA amplification and iii) final read sampling by sequencing. In the right we show variables that model quantities at each step.

Number of fragments f_t. In the absence of biases, transcript positions have homogeneous fragmentation probabilities. Thus ideally f_t should follow a Poisson distribution $f_t \sim \mathcal{P}(\frac{x_t}{a})$. The ratio $\frac{x_t}{a}$ represents the expected count at position t corrected by the

mean amplification. We account for possible overdispersion induced by the first steps of the protocol (fragmentation, RT, size selection) by using a random scaling factor s_t of mean 1. For simplicity, we choose a Gamma distribution for s_t with shape $\frac{1}{k_s}$ and write the distribution of f_t as

$$\begin{aligned} f_t \mid x_t, s_t &\sim \mathcal{P}(\frac{x_t}{a} \cdot s_t) \\ f_t \mid x_t &\sim \mathcal{NB}(\frac{x_t}{a}, \kappa_s). \end{aligned} \tag{3.1}$$

In the datasets we included in our study it seems that the counts within a transcript tend to cluster in islands with relatively homogeneous coverage. These islands (outlined by grey bars in figure 3.4) are of range 5-50 bp and are often separated by segments with very low counts. Though we could relate this local correlation to various RNA-Seq steps, we choose to consider it as anterior to amplification. Our choice is based on the intuition that the local correlation might be due to specific regions having a higher fragmentation or RT initiation probability. We include the local correlation of molecule scaling by taking s_t as a piece wise constant variable $\pi(s_t \mid s_{t-1}) = \alpha_s \cdot \delta_{s_{t-1}}(s_t) + (1 - \alpha_s) \cdot \Gamma(s_t; 1, \frac{1}{\kappa_s})$, where the $\frac{1}{1-\alpha_s}$ is the average correlation window length.

Amplification. PCR amplification and depth of sequencing scale the number of fragments prior to their identification. Each fragment is multiplied according to a random distribution of mean a (we take a Gamma for convenience). The amplification coefficient for fragments with $5' - end$ in position t is

$$a_t \sim \Gamma(a, \kappa), \tag{3.2}$$

where a_t is the expected number of sequenced reads for an initial cDNA fragment.

Final read sampling. The count of reads aligned to the position t (y_t) comes from a simple Poisson distribution with mean equal to the abundance of amplified fragments

$$y_t \mid f_t, a_t \sim \mathcal{P}(f_t \cdot a_t). \tag{3.3}$$

For the estimation of amplification parameters we pay a particular attention to isolated counts, i.e. positions with positive counts in regions with close to zero average count. In some datasets (e.g. *E. coli* 2 in figure 3.7) we observed a high number of isolated counts

with with values at 1 while simultaneously having high amplification coefficients. This excess possibly represents non amplified reads. We add a probability p_1 to have counts of 1. This term is important only for the isolated counts and thus we will not consider it in writing the statistics for read distribution within ORFs. Also, we do not consider it in the Parseq emission model (section 4.1.1), the distribution of isolated reads being accounted by a more general background distribution.

From the equations 3.1, 3.2 and 3.3 we can write the distribution of counts for a given expression level x_t and local correlation s_t as

$$y_t \mid x_t, s_t \quad = \quad \sum_{f_t=0}^{\infty} \mathcal{P}(f_t; \frac{x_t s_t}{a}) \cdot \mathcal{NB}(y_t; f_t \cdot a, \kappa). \tag{3.4}$$

3.3 Estimation of Read Count Model Parameters

For practical reasons we decided to estimate the parameters of the read count model (*i.e.* κ, a, κ_s, α_s and p_1) directly from the characteristics of the read count distribution within transcripts without using the expression profile reconstruction.

The estimation relies on the distribution of isolated counts and on the distribution of reads within ORFs. For the later we consider the relationships between mean and variance, mean and the fraction of zero-counts (frequency of positions with a count equal to zero) and the autocorrelation of counts. Most of the parameters play a role in all these relationships and we decide to estimate them simultaneously.

3.3.1 Relationships between Parameters and Count Distribution

In **isolated counts**, i.e. positive read counts occurring in regions with very low coverage, for a position t, the observed counts are very likely issued from a single initial molecule ($f_t = 1$) because x_t is very small. Thus the distribution of these positions is shaped only by amplification. From isolated counts we derive relations for the parameters p_1, α and κ. This distribution $\pi(y_t \mid f_t = 1, y_t > 0) =: \pi_{is}(y_t)$ writes

$$\pi_{is}(y_t) = (1 - p_1) \cdot \frac{\mathcal{NB}(y_t; a, \kappa)}{1 - \left(\frac{1}{a\kappa+1}\right)^{\frac{1}{\kappa}}} \cdot \mathbf{1}_{\{y_t \geq 1\}} + p_1 \cdot \mathbf{1}_{\{y_t = 1\}}. \tag{3.5}$$

From the **ORF count distribution** we use relationships between count variance and mean, and zero-counts and mean, to derive relations for amplification parameters and scaling of fragments. From the equation 3.4, after integration over the possible values of s_t and assuming the autocorrelation effects are mitigated, the density of the marginal distribution $y_t \mid x_t$ writes

$$\pi(y_t \mid x_t) = \sum_{f_t \geq 0} \mathcal{NB}(f_t; \frac{x_t}{a}, \kappa_s) \cdot \mathcal{NB}(y_t; f_t a, \kappa). \tag{3.6}$$

Using the law of total variance and writing the probability of zero-counts we obtain

$$\mathbb{V}(y_t \mid x_t) = (\kappa + \kappa_s + \kappa_s\kappa) \cdot x_t^2 + x_t \cdot (1 + a\kappa + a), \tag{3.7}$$

$$\mathbb{P}(y_t = 0 \mid x_t) = \sum_{f_t \geq 0} \mathcal{NB}(f_t; a, \kappa_s) \cdot \mathcal{NB}(0; f_t a, \kappa). \tag{3.8}$$

We draw the attention of the reader that the variance expression $\mathbb{V}(y_t \mid x_t)$ changes in the case of missing amplification ($k = 0, a = 0$) to $\mathbb{V}(y_t \mid x_t) = (\kappa_s) \cdot x_t^2 + x_t$ (the variance of a NB).

Count correlations is induced in our model by the scaling variable s_t. For an ORF g and a lag ℓ the count autocorrelation is derived from the transition kernel of s_t and writes

$$\text{cor}(y_{t+\ell}, y_t \mid x_g) = \frac{x_g^2 \cdot \kappa_s}{\sigma_g^2} \cdot \alpha_s^\ell, \tag{3.9}$$

where σ_g^2 is the variance of the counts in ORF g and can be approximated empirically or derived from equation 3.7. In practice we compute it empirically.

We aim to fit the distribution of $\hat{\text{cor}}_{1^\ell}$, the estimator of the lag 1 correlation ($\text{cor}(y_{t+1}, y_t)$) obtained from the lag ℓ empirical correlation. We compute it by averaging the lag ℓ count autocorrelation over the set of ORFs (G) and then correcting for the lag by taking the ℓ root: $\hat{\text{cor}}_{1^\ell} = [\frac{1}{G} \sum_{g=1:G} \text{cor}_g(y_{t+\ell}, y_t)]^{1/\ell}$.

3.3.2 Read Count Model Parameter Optimisation

For the simultaneous estimation of parameters we need to find comparable statistics between the four data-sources we use (the distribution of isolated counts and the three indicators of the distribution of counts within ORFs).

The coefficient of determination R^2 can be compared between heterogeneous datasets and permits the optimisation of the complete set of parameters. This coefficient indicates how well a statistical model fits the data-set and is computed as the ratio between the sum of squares of residuals and the total sum of squares

$$1 - R^2 = \frac{\sum_k (v_k - \tilde{v}_k)^2}{\sum_k (v_k - \bar{v})^2},$$

where $\tilde{v}_k$ is the model predicted value, v_k is the observed data, and $\bar{v} = \frac{1}{K} \sum_{k=1:K} v_k$. A coefficient $R^2 = 1$ means the predictions $\tilde{v}$ perfectly fit the data ($R^2 = 0$ or $R^2 < 0$ indicate that the predictions are unrelated to the observed data). We denote θ the vector of parameters $\{p_1, \kappa, a, \kappa_s, \alpha_s\}$ and compute the coefficients of determination for each data source.

For the isolated counts, we write the coefficient R^2 based on the histogram of the empirical density $\pi_{is}(y)$ for $y \geq 1$

$$1 - R^2_{is}(\theta) = \frac{\sum_y (\hat{\pi}_{is}(y) - \pi_{is}(y;\theta))^2}{\sum_y (\hat{\pi}_{is}(y) - \bar{\pi}_{is}(y))^2}. \tag{3.10}$$

For the zero-counts and variance versus mean relationships we write the R^2 coefficients

$$1 - R^2_{var}(\theta) = \frac{\sum_g (\mathbb{V}_\theta(y_t \mid x_g) - \sigma^2_g)^2}{\sum_g (\sigma^2_g - E_g(\sigma^2_g))^2}, \tag{3.11}$$

$$1 - R^2_{zero}(\theta) = \frac{\sum_g [(\mathbb{P}_\theta(y_t = 0 \mid x_g) - z^0_g)^2}{\sum_g (z^0_g - E_g(z^0_g))^2}, \tag{3.12}$$

where we use $g = 1 : G$ for the index of an ORF index with empirical mean of the read counts x_g, empirical variance of the reads counts σ^2_g, and empirical fraction of zero-counts z^0_g.

The R^2 coefficient for the count correlation aims to fit the distribution of the lag 1 correlation ($\text{cor}(y_{t+1}, y_t)$) obtained empirically from the lag ℓ empirical correlation and writes

$$1 - R^2_{cor}(\theta) = \frac{\sum_l (\text{cor}_{\theta_l}^{1/l} - \hat{\text{cor}}_{1^\ell})^2)}{\sum_l (\hat{\text{cor}}_{1^\ell} - \frac{1}{L} \sum_{l=1:L} \hat{\text{cor}}_{1^\ell})^2}. \tag{3.13}$$

The parameters a, κ, κ_s, α_s and p_1 are obtained by the joint optimisation on θ of equations 3.10, 3.11 ,3.12 and 3.13: $R^2_{tot}(\theta) = R^2_{is}(\theta) + R^2_{var}(\theta) + R^2_{zero}(\theta) + R^2_{cor}(\theta)$.

We draw an uncertainty region for these parameters by building a Markov chain constrained to values close to the optimized $\tilde{R}^2_{tot} = \max_\theta R^2_{tot}(\theta)$. We propose Gaussian distributed moves with variance σ^2 and accept values with $R^2_{tot} > 0.8 \cdot \tilde{R}^2_{tot}$. For a parameter θ the acceptance probability of a proposed θ^* value at iteration $(n+1)$ writes

$$\min\left(1, \mathbf{1}_{\{R^2_{\theta^*} > 0.8 \cdot \tilde{R}^2_{tot}\}}(\theta^*) \frac{\mathcal{N}(\theta^{(n)}; \theta^*, \sigma)}{\mathcal{N}(\theta^*; \theta^{(n)}, \sigma)}\right).$$

3.3.3 Practical Implementation of the Density with the New Count Model

The new density (equation 3.4) is computational intensive as it requires summing over the possible values of f_t. We tackle this by pre-computing for the optimized set of parameters the density values for the full range of observed counts and for a discrete set of values of

the product $x_t \cdot s_t$ and of f_t. In practice we set the values of $x_t \cdot s_t$ within the range of probable values $(0, 10 \cdot \max_g x_g)$, where we remind that x_g is the count expectation for the ORF g. During pre-computation, for each value of $x_t \cdot s_t$ we sum over a set of f_t values for which the Poisson probability $\mathcal{P}(f_t; \frac{x_t s_t}{a})$ is more than 10^{-6}.

To compute the density for given values of y_t and $x_t \cdot s_t$ we perform linear approximations using the closest grid values.

3.4 Parameter Values for the Read Count Model

The distribution of isolated counts and the identification of homogeneous expression segments are required for the parameter estimation (procedure described in section 3.3). In practice we use the following criteria to define these distributions:

- isolated counts: positions with positive counts and less than 4 additional reads in a 400 bp window;
- homogeneous expression segments: 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. 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 (see figure 3.2).

Within ORFs we encounter two autocorrelation effects: small islands (5-50 bp) of very correlated counts and long range correlation due to coverage smooth progressive changes. Because we focus on the short range correlation we perform the estimation of autocorrelation on relatively short ORFs (200 -1000 bp) and for correlation windows between 1 and 20.

In figure 3.7 we illustrate the estimation procedure. For each data-set R^2 coefficients are maximized simultaneously for the distributions of isolated counts, variance and zero counts and autocorrelation. Interestingly we observe that the datasets that we used for our first analysis (and for which we will present results on transcription boundaries estimation) show the highest levels of amplification (*E. coli* 2-3, *S. cerevisiae* 1). For these datasets, a NB model does not fit well the variance vs . mean and zero-counts vs. mean relationships. Other data-sets, like *E. coli* 1, show a perfect fit with the NB and very reduced amplification ($a \approx 0$ and $k_s \approx 0.2$). The amount of autocorrelation differs between data-sets . While we can have practically no autocorrelation ($\alpha_s \approx 0.1$ and therefore the average autocorrelation window size $w_s = \frac{1}{1-\alpha_s}$ is ≈ 1 for *S. cerevisiae* 1), some datasets have long local correlation windows (size 14 for *C. albicans* 1). In the data-sets *C. albicans* 1 and *E. coli* 1 we also notice the effect of the correlation induced by smooth progressive changes (grey histogram).

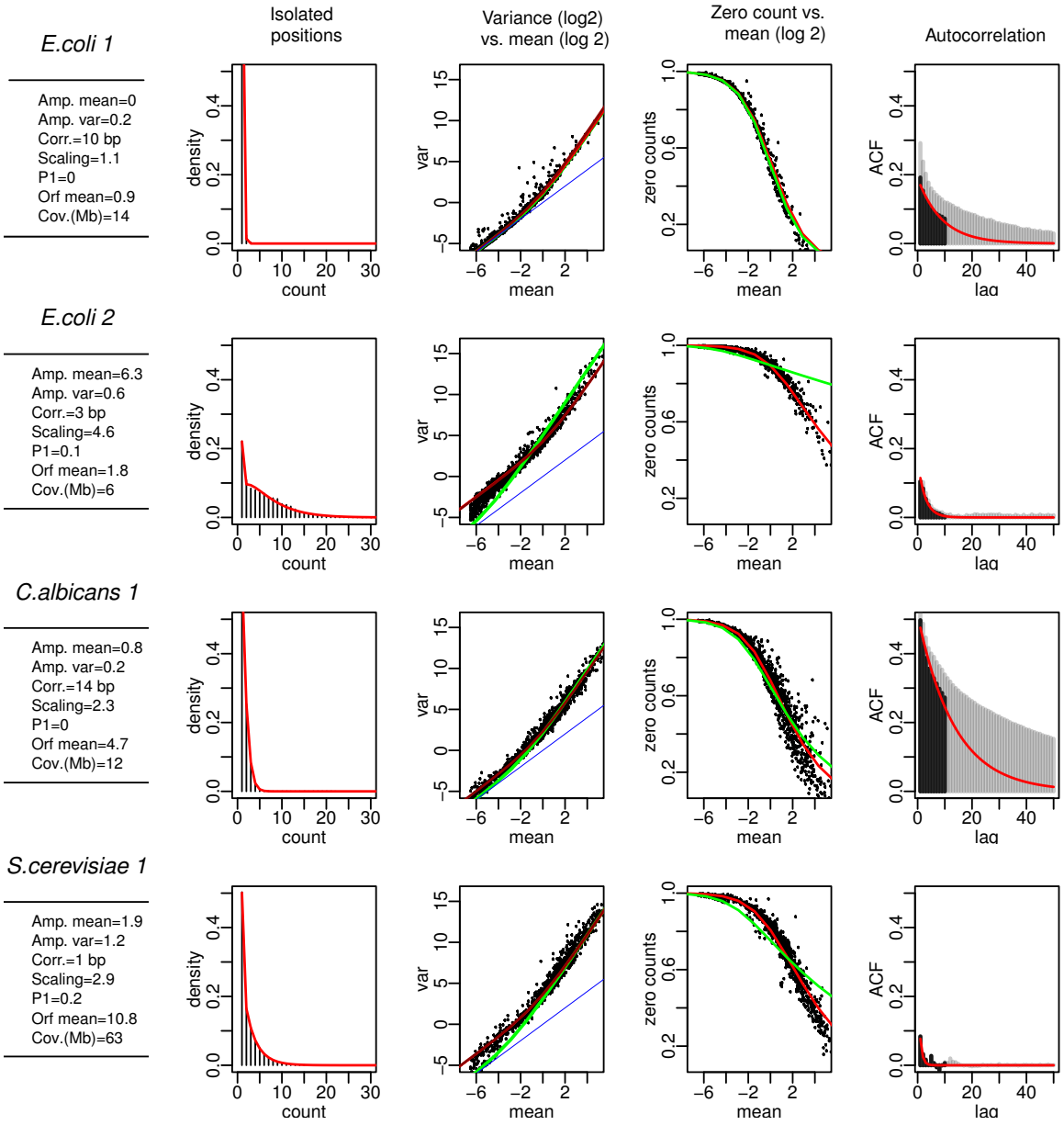

Figure 3.7: Estimation of the read counts parameters (a, κ, κ_s, α and p_1) for data-sets *E. coli* 1-2, C.*albicans* 1 and S.*cerevisiae* 1. Right plots: distribution of isolated reads (reads with positive counts and less than 4 additional reads in a 400 bp window). Middle plots: variance and zero counts distribution vs mean for ORFs. Each point represents an ORF. Left: autocorrelation of counts in ORFs; black bars represent autocorrelation for small and average length ORFs (200-1000 bp); gray bars autocorrelation for long ORFs ($\geq$1000 bp). Red line: mixture model estimation; Green line: NB estimation; Blue line: Poisson estimation.

In figure 3.8 we show estimation results on 16 data-sets. As expected, parameter values for datasets coming from the same run or study are very similar (*E. coli* 2-3, *C. albicans* 2-3 and *F. psychrophilum* 1-2). We find that most values for the autocorrelation window are in the interval $(4, 8)$. High autocorrelation window sizes (w_s) coupled with high scaling (κ_s) might pose problems in disentangling transcription breakpoints from local shifts in coverage induced by bias. Several data-sets show low amplification values. For these, the estimation can be done by approximating the emission model with a NB distribution. Finally, the low degree of parameter correlation, and notably the low correlation between κ and κ_s, indicates that the read count distribution is affected by several independent factors.

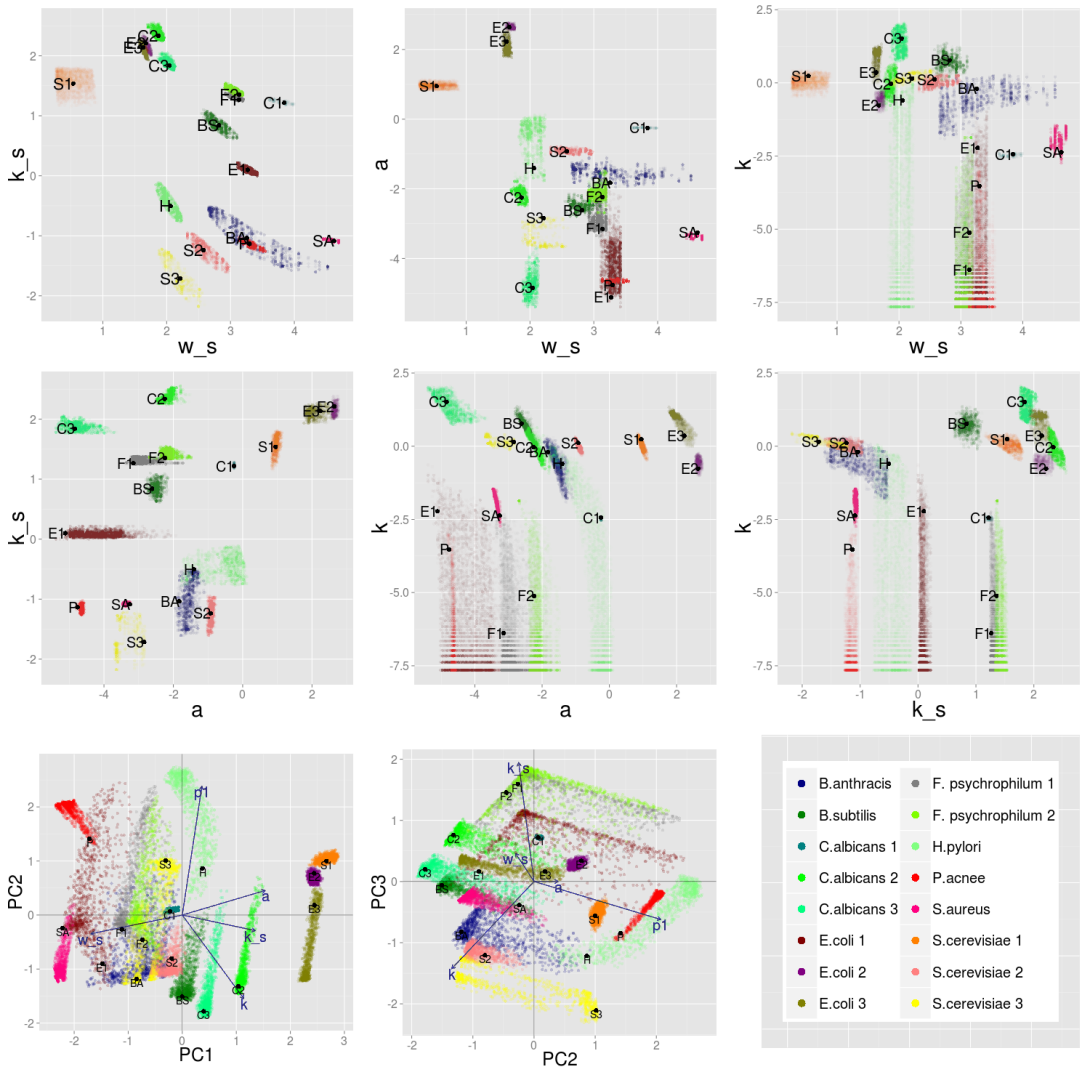

Figure 3.8: Landscape of parameter values (a, κ, κ_s, $w_s = \frac{1}{1-\alpha_s}$) obtained for the count mechanistic model. We show $\mathbf{log_2}$ values of the parameters for several datasets from yeast and bacteria. We perform pairwise investigations (frames 1-6) and a PCA analysis for the first 3 components (frames 7 and 8). Each color corresponds to a data-set, the legend is shown in frame 9. For each parameter we show a "close to optimal" interval (i.e. for which $R^2 \geq 0.8 \cdot \tilde{R}^2_{tot}$). The optimal value is shown by a black dot.

Chapter 4

Parseq: from read counts to the transcription profile

We present in this chapter our method, named Parseq, for the reconstruction of the transcription landscape. We design a State Space Model where the observations are the read counts and the latent variable the expression level. The emission, i.e. dependency between observations and latent variables is built upon the new count emission model. The transition kernel for the expression level is a mixture model that accounts for the possibility of not changing the level and of smooth or significant level changes. Transcriptional landscape reconstruction is then conducted with a Sequential Monte Carlo method, the Particle Gibbs. We show here procedures to estimate the model parameters and to approximate transcription levels at base-pair resolution.

We include in this chapter several results concerning the application of Monte Carlo methods. We demonstrate empirically Particle Gibbs improvements in exactness over Forward filtering using a validation method of our design. This method can also be used to capture implementation errors. Next, we present a study on performances increase by using different proposal functions and we adapt a proposal tuning algorithm. Parseq results were evaluated on two datasets from *S. cerevisiae* and *E. coli*. We compare calling of transcribed positions and transcription breakpoints estimation with two other methods: Cufflinks and Rockhopper.

4.1 A SSM for Transcription Levels and Read Counts

We summarize the multiple layers of variables involed in our probabilistic model and the conditional dependence between the variables through a Directed Acyclic Graph (DAG) in figure 4.1.

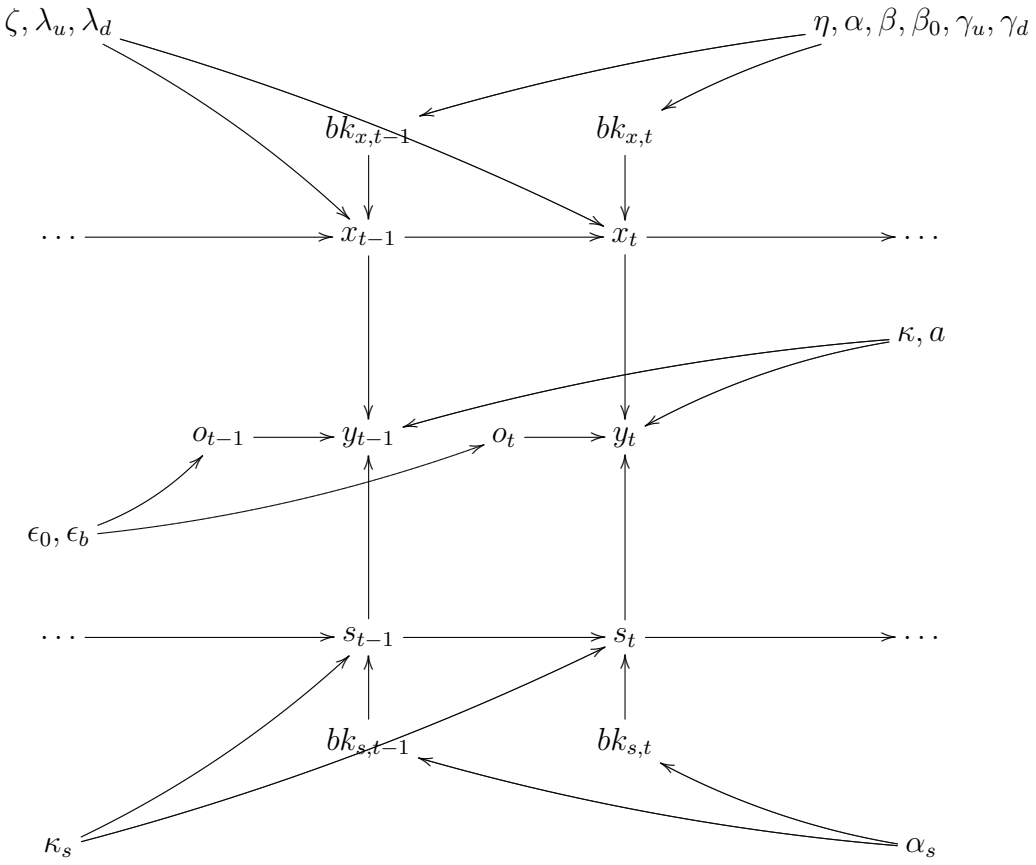

Figure 4.1: The DAG of the Parseq model. Main variables: $\mathbf{y} = (y_t)_{t\geq 1}$, the sequence of observations (read counts); $\mathbf{x} = (x_t)_{t\geq 1}$, the latent expression level, and $\mathbf{s} = (s_t)_{t\geq 1}$, the local scaling/correlation variable. Auxiliary variables have been added to ease the design of the MCMC algorithm: $\mathbf{o} = (o_t)_{t\geq 1}$ indicating if the observation has been generated by the outlier model; $\mathbf{bk_x} = (bk_{x,t})_{t\geq 1}$ and $\mathbf{bk_s} = (bk_{s,t})_{t\geq 1}$ indicating the type of transition between adjacent positions on $\mathbf{x}$ (shift, drift, no-change) and on $\mathbf{s}$. Parameters are inserted in the DAG by curved arrows. Emission variables that were integrated out in the final formula (a_t, f_t), are not shown.

Of note, transition kernels for $\mathbf{x}$ 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 for these mixtures in order to facilitate the design of the MCMC.

4.1.1 Emission Model for the Read Counts

In the section 3.2 we describe a mechanistic model for the RNA-Seq counts $\mathbf{y}$ distribution. We remind that y_t corresponds to the number of reads with 5'-ends mapping at position t. Given the number of fragments sampled before the PCR f_t, and the amplification coefficient a_t, the distribution of the observed read count for a position t is a Poisson distribution $\mathcal{P}(f_t \cdot a_t)$. In the emission model we want to consider also cases where the reads at a specific position originate from technical errors. Thus we introduce two emission components that model the background noise and outliers. The first wants to account for low isolated counts that might be the product of pervasive transcription, misalignment and contaminating DNA. We assume that the background noise consists in reads with fragment counts f_t of 1 and they have a zero-truncated Poisson distribution with mean a_t. The outliers component aims mainly to capture amplification errors (high isolated counts). We assume outliers have an uniform distribution on the count distribution range. Thus, the emission density writes as a mixture

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

where $\varepsilon, \varepsilon_b, \varepsilon_o$ represent the probabilities of for 'normal', background and outlier emission.

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

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

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

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

  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 $\mathcal{NB}_{-\{0\}}(a, \kappa)$.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'.

4.1.2 Longitudinal Model of Transcriptional Level

The transcription landscape requires a state space that distinguishes expressed ($x_t > 0$) and non-expressed ($x_t = 0$). The Markov transition kernel $k(x_t; x_{t-1})$ needs to reflect the probability of change between these states. In this section we will describe such a kernel that tries to mimic the transcription changes. The simplest mixture kernel writes

$$\mathbf{1}_{\{x_{t-1}=0\}} \left[(1-\eta)\,\delta_0(x_t) + \eta\, f(x_t)\right] + \mathbf{1}_{\{x_{t-1}>0\}} \left[(1-\beta_0)\, g(x_t; x_{t-1}) + \beta_0\, \delta_0(x_t)\right],$$

where η and β_0 are probabilities for expression status change, f and g are generic densities for $x_t > 0$, $\mathbf{1}$ denotes the indicator function that serves to indicate whether $t-1$ is an expressed or non-expressed position and δ_0 denotes the Dirac delta function with mass at point 0 that gives a non-zero probability for regions with 0 transcription level.

This simple model hinders the complete description of expression level changes. Following a similar work on tiling array data [Nicolas et al., 2009], transitions within transcribed regions (incorporated in g) further subdivide into three types. We account for unchanged transcription level and for changes that differ by their amplitudes and are referred as shifts (large amplitude) and drifts (small amplitude).

The shifts correspond to changes of transcription level within transcribed regions – accounting for transcription initiation and termination sites in presence of overlapping transcription units. The coexistence of shifts and drifts is designed to pull apart well defined initiation or termination sites internal to transcribed regions from smoother changes in measured transcriptional levels that can have a biological origin (e.g., random termination events) or can reflect technical artefacts (e.g., longitudinal bias caused by mRNA capture and fragmentation protocols).

Our complete Markov transition kernel $k(x_t; x_{t-1})$ for transcriptional level writes

$$\begin{aligned}\mathbf{1}_{\{x_{t-1}=0\}} \quad &\cdot \quad \left[(1-\eta)\delta_0(x_t) + \eta f(x_t)\right] + \\ \mathbf{1}_{\{x_{t-1}>0\}} \quad &\cdot \quad \left[\alpha\delta_{x_{t-1}}(x_t) + \beta f(x_t) + \beta_0\delta_0(x_t) + \gamma_u g_u(x_t; x_{t-1}) + \gamma_d g_d(x_t; x_{t-1})\right],\end{aligned}$$

where the parameters $\eta \in (0,1)$ and $(\alpha, \beta, \beta_0, \gamma_u, \gamma_d) \in (0,1)^5$ with $\alpha+\beta+\beta_0+\gamma_u+\gamma_d = 1$ define the probabilities of the different types of moves. The terms $f(x_t)$, $g_u(x_t; x_{t-1})$ and $g_d(x_t; x_{t-1})$ are probability densities for the transcription level x_t: at the beginning of a transcribed region or after a shift ; after an upward drift , and after a downward drift , respectively.

Understanding the kernel mixture is made easier by looking at each component and by recording the type of the change between positions $t-1$ and t in the auxiliary variable $bk_{x,t}$:

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

– The expression level changes to a value independent of the previous position with probability β. In this case $bk_{x,t}$ = 'shift' and the density of x_t is drawn from an exponentional distribution with rate ζ (density $\zeta \cdot e^{-x_t \cdot \zeta}$).

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

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

$$k_s(s_t; s_{t-1}) \quad = \quad \alpha_s \cdot \delta_{s_{t-1}}(s_t) + (1 - \alpha_s) \cdot \Gamma(s_t; 1, \frac{1}{\kappa_s}) \,. \tag{4.1}$$

This corresponds to a piecewise constant Gamma (parametrisation in 3.0.2) 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 $t-1$ and t is α_s and the type of move ('move' or 'no-move') is recorded in $bk_{s,t}$.

Transcript borders are detected on the basis of significant changes in read counts. Therefore, high variability in read counts can lead to breakpoint over-predictions resulting in a loss of specificity when not properly incorporated in the model. We palliated this need by introducing the two different components in our model: the drift term on the transition kernel for progressive variations as opposed to the abrupt changes modeled by shifts, and the local scaling Markov-dependent variable **s** intended to capture short-range autocorrelations. We analysed models that don not account for the drift moves or local correlations and we illustrate this on the data-set *S. cerevisiae* 1 (see results in table 4.3).

4.2 Reconstruction of the Transcription Landscape

In this section we will describe how the recent SMC developments discussed in chapter 2 can provide an exact solution to approximating the expression level profile. First, we show a PG complete algorithm that aims at estimating the expression profile and at disentangling short-range correlations due to local bias and long range correlations that should have a biological meaning. This MCMC algorithm combines filtering done through CSMC and smoothing done through Backward sampling for trajectory update and sampling from the posterior distribution for parameter update. The CSMC results in highly correlated trajectories and, with a view to improve mixing, we design an additional step in the Gibbs algorithm. Besides updating $x \mid s, \Theta, y$ and $s \mid x, \Theta, y$ we mitigate the negative correlation of x and s by introducing a new update $x \cdot s \mid \Theta, y$. While this step improves mixing it has an additional computation cost. We take also a block update option. Initially this was mainly a recourse against the inertia induced by coalescence in CSMC approximations of long sequences. Backward sampling greatly improves mixing and does not require the splitting of long sequences. We mantain the block update with longer block sizes for memory efficiency.

The implementation of complex MCMC algorithms such as the ones we use is error prone and might lead to errors in parameter and trajectory approximation hard to detect. In the same time the correctness of results might be influenced by factors intrinsic to the method (e.g. several SMC methods provide accurate estimations only asymptotically). We implement a strategy to verify the algorithm accuracy. We discuss next the choice of the instrumental function. This is of high importance for Monte Carlo estimations and we present the construction of our expression level proposal and a simple adaptive Monte Carlo method for the optimisation of weights in a mixture proposal.

The complete Parseq workflow for estimating breakpoints and reconstructing transcripts requires the identification of positions with high breakpoint probability. We implement a post-processing step of the Parseq results that uses local score procedures for detecting regions with high breakpoint probability and for assembling transcription continuous units. We end this section with an evaluation of breakpoint estimation on *S. cerevisiae* 1 and *E. coli* 2. We use for comparison purposes the results obtained from two popular algorithms: Cufflinks and Rockhopper.

4.2.1 Particle Gibbs for Expression Level Reconstruction

We implement a Particle Gibbs algorithm aiming to sample the joint distribution

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

Each PG sweep ($n = 1 : N$) consists of updating one or a subset of the variables ($\mathbf{x}$, $\mathbf{s}$ and the parameters) according to their conditional distribution:

- Update $\mathbf{x}$ preserving $\mathbf{x} \mid \mathbf{s}, \Theta, \mathbf{y}$ with CSMC by blocks (algorithm 11).
- Update $\mathbf{s}$ preserving $\mathbf{s} \mid \mathbf{x}, \Theta, \mathbf{y}$ with CSMC by blocks (similar to algorithm 11).
- Simultaneous $\mathbf{x}$ and $\mathbf{s}$ update preserving $\mathbf{x}, \mathbf{s} \mid \Theta, \mathbf{y}$ with CSMC by blocks (alg. 13)
- Update of $\mathbf{o}$ according to $\mathbf{o} \mid \mathbf{x}, \mathbf{s}, \Theta, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of $\mathbf{bk_x}$ according to $\mathbf{bk_x} \mid \mathbf{s}, \mathbf{x}, \mathbf{o}, \Theta, \mathbf{y}$ (the variables written in gray 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 $\mathbf{bk_s} \mid \mathbf{s}, \mathbf{x}, \mathbf{o}, \mathbf{bk_x}, \Theta, \mathbf{y}$. 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 $\alpha, \gamma_u, \gamma_d, \beta, \beta_0 \mid \mathbf{s}, \mathbf{x}, \mathbf{o}, \mathbf{bk_x}, \mathbf{bk_s}, \eta, \zeta, \epsilon_o, \epsilon_b, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of η according to $\eta \mid \mathbf{s}, \mathbf{x}, \mathbf{o}, \mathbf{bk_u}, \mathbf{bk_s}, \alpha, \gamma_u, \gamma_d, \beta, \beta_0, \zeta, \epsilon_o, \epsilon_b, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of ζ according to $\zeta \mid \mathbf{s}, \mathbf{x}, \mathbf{o}, \mathbf{bk_x}, \mathbf{bk_s}, \alpha, \gamma_u, \gamma_d, \beta, \beta_0, \eta, \epsilon_o, \epsilon_b, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).
- Update of (ϵ_o, ϵ_b) according to $\epsilon_o, \epsilon_b \mid \mathbf{s}, \mathbf{x}, \mathbf{o}, \mathbf{bk_x}, \mathbf{bk_s}, \alpha, \gamma_u, \gamma_d, \beta, \beta_0, \eta, \zeta, \mathbf{y}$. This conditional distribution is sampled directly (Gibbs-type update).

We will discuss in detail the first step, i.e. the CSMC update of $x_{1:T}$ preserving $x_{1:T} \mid s_{1:T}, \Theta, y_{1:T}$. In our context an additional difficulty comes from the length of the $1 : T$ sequence which for filtering algorithms leads to poor mixing for any reasonable number of particles and that also might pose memory issues in a straight implementation.

4.2.2 The Block Update Conditional SMC

We implemented a block update version of the Particle Gibbs algorithm to circumvent these problems. Of note, our piecewise constant models for x 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.

Algorithm 11 The block update Conditional SMC

Objective: sample $x^{(n)}_{t_0:t_1} \mid x^{(n-1)}_{1:T}, \Theta^{(n-1)}, y_{1:T}$ for any $0 \leq t_0 < t_1 \leq T$.

1. For $t = t_0$,
   (a) For the first $P-1$ particles $1 \leq p < P$, draw $x^p_{t_0}$ from the proposal $q_{x,t_0}(x^p_{t_0}; x^{(n-1)}_{t_0-1})$. Set $x^N_{t_0} = x^{(n-1)}_{t_0}$
   (b) For $1 \leq p \leq P$, compute the particle weights $w^p_{t_0} = \frac{k_x(x^p_{t_0}; x^{(n-1)}_{t_0-1}) \cdot e(y_{t_0}; x^p_{t_0}, s_{t_0})}{q_{x,t_0}(x^p_{t_0}; x^{(n-1)}_{t_0-1})}$
   Compute $W^p_{t_0} = \frac{w^p_{t_0}}{\sum_{p=1:P} w^p_{t_0}}$
2. From $t = t_0 + 1$ to $t = t_1$,
   (a) If $1/\sum_{p=1:P}(W^p_{t-1})^2 > P/4$ (ESS $> P/4$) then $a^p_t = p$ for $1 \leq p \leq P$.
   (b) If $1/\sum_{p=1:P}(W^p_{t-1})^2 \leq P/4$ then for $p = 1 : P-1$ draw index of the ancestor particle a^p_t from weights $(W^p_{t-1})_{1:P}$. Set $a^P_t = P$ and $(W^p_{t-1})_{1:P} = \frac{1}{P}$.
   (c) For $p = 1 : P-1$, draw x^p_t from the proposal density $q_{x,t}(x^p_t; x^{a^p_t}_{t-1})$. Set $x^P_t = x_t$.
   (d) For $p = 1 : P$ compute particle weights $w^p_t = W^{a^p_t}_{t-1} \cdot \frac{k_x(x^p_t; x^{a^p_t}_{t-1}) \cdot e(y_t; x^p_t, s_t)}{q_{x,t}(x^p_t; x^{a^p_t}_{t-1})}$
   Compute $W^p_t = \frac{w^p_t}{\sum_{p=1:P} w^p_t}$
3. For $t = t_1$,
   (a) Each particle p has a probability proportional to $w^p_{t_1} \cdot k_x(x^{(n-1)}_{t_1+1}; x^p_{t_1})$.
   (b) Backward sample (algorithm 12) $x^{(n)}_{t_0:t_1}$ from $\left(x^p_{t_0:t_1}\right)_{p=1:P}$ with filtering weights $\left(w^p_{t_0:t_1}\right)_{p=1:P}$

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_x and q_x.

If $x_{t_1+1} > 0$ the probability of proposing x_{t_1+1} is zero. Thus, all the new trajectories created by this algorithm will have a breakpoint between position t_1 and $t_1 + 1$. Due to the choice of the, which includes the mass $\delta_{\{x_{t-1}\}}(x_t)$, we have a non zero probability of a piecewise segment $x_{t_1:t_1+1}$. This imposes some constraints on the choice of the block $x_{t_0:t_1}$

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 ($x_{t_1+1} = 0$).

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 $t'_1 = t_0 + \ell$. For algorithms without backward sampling 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. Backward sampling algorithms allow us to deal with longer sequences and we set $\ell = 10000$ to limit the memory usage. 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 $x_t = 0$ 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).

Initially we used a backtrack strategy to reconstruct the trajectory $x^*_{t_0:t_1}$ from the filtering generated sample $\left(x^p_{t_0:t_1}\right)_{p=1:P}$ using the particle ancestor index $\left(a^p_{t_0:t_1}\right)_{p=1:P}$: for each $t = t_1 : t_0$ set $p^\star_{t-1} = a^{p^\star_t}_t$ and $x^\star_{t-1} = x^{p^\star_{t-1}}_{t-1}$. To improve mixing we implement a Backward sampling (algorithm 3) that accounts for the aforementioned Parseq kernel specificities, namely that $k(x_t \mid x_{t-1})$ is a mixture including a Dirac mass in x_{t-1} (algorithm 12).

Algorithm 12 Backward sampling

Objective: backward sample a trajectory $x^*_{t_0:t_1}$ from $t = t_1$ to $t = t_0$ from a filtering generated sample $\left(x^p_{t_0:t_1}\right)_{p=1:P}$ with filtering weights $\left(w^p_{t_0:t_1}\right)_{p=1:P}$

1. Draw index $p^\star_{t_1}$ and set $x^\star_{t_1} = x^{p^\star_{t_1}}_{t_1}$
2. For $t = t_1 : t_0 + 1$ do:
   (a) if $x^{a^{p^\star_t}_t}_{t-1} \neq x^\star_t$ then draw the $p^\star_{t-1}$ index from $P(p^\star_{t-1} = p) \propto w^p_{t-1|t}$ where weights $w^p_{t-1|t} = k(x^\star_t \mid x^p_{t-1}) \cdot w^p_{t-1}$. Set $x^\star_{t-1} = x^{p^\star_{t-1}}_{t-1}$
   (b) else $p^\star_{t-1} = a^{p^\star_t}_t$ and $x^\star_{t-1} = x^{a^{p^\star_t}_t}_{t-1}$

4.2.3 Particle Gibbs with Simultaneous Update of x and s

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

This motivated the development of another block update Particle Gibbs algorithm that can update $\mathbf{x}$ while preserving as much as possible the product $x_t \cdot s_t$ by applying simultaneous modifications to $\mathbf{s}$ (algorithm 13). For the sake of clarity let z_t denote the product $x_t \cdot s_t$. If $x_t > 0$ the model can be rewritten in terms of $\mathbf{x}$ and $\mathbf{z}$ instead of $\mathbf{x}$ and $\mathbf{s}$; the trajectory of $\mathbf{z}$ being fully determined by the trajectory of $\mathbf{x}$ and the values of z_t at breakpoint positions $\{t : bk_{s,t} = \text{'change'}\}$. The Conditional SMC update described below targets $\pi(x_{t_0:t_1} \mid x_{t<t_0}, x_{t>t_1}, z_{t_0}, z_{\{t_0<t\leq t_1 : bk_{s,t}=\text{'change'}\}}, \Theta, x_{t_0:t_1} > 0, y_{1:T})$. The new values of $\mathbf{s}$ are obtained after updating $\mathbf{x}$ when reverting to the original parametrization of the model in terms of $\mathbf{x}$ and $\mathbf{s}$, by using the relation $s_t = z_t/x_t$ at breakpoint positions $\{t : bk_{s,t} = \text{'change'}\}$.

The procedure to select a block $x_{t_0:t_1}$ consists of: selecting randomly t'_0, searching for $t_0 = \min\{t \geq t'_0 : x_{t_0} > 0\}$, setting $t'_1 = t_0 + l$ where l is a random variable with mean $\min\{5/(1-\alpha), P\}$ (see subsection 4.2.2), and then searching for $t_1 = \max\{t_1 \leq t'_1 : x_{t0:t1} > 0\}$. 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.

Algorithm 13 Conditional SMC for simultaneous update of $\mathbf{x}$ and $\mathbf{s}$

Objective: sample $x_{t_0:t_1}s_{t_0:t_1} \mid x_{t<t_0}, x_{t>t_1}, z_{t_0}, z_{\{t_0<t\leq t_1:bk_{s,t}=\text{'change'}\}}, \Theta, x_{t_0:t_1} > 0, y_{1:T}$.

1. For $t = t_0$,

   (a) For $p = 1 : P-1$ draw x_t^p from the $q_{x,t_0}(x_{t_0}^p; x_{t_0-1}^{(n-1)})$. Set $x_{t_0}^P = x_{t_0}^{(n-1)}$

   (b) For $p = 1 : P$, set $s_{t_0}^p = z_{t_0}/x_{t_0}^p$ where $z_t = x_t s_t$

   (c) For $p = 1 : P$, compute particle weights

   $$w_{t_0}^p = \frac{k_x(x_{t_0}^p; x_{t_0-1}^{(n-1)}) \cdot e(y_{t_0}; z_{t_0}^p) \cdot \mathbf{1}_{\{x_{t_0}^p>0\}} \cdot \pi_s(s_{t_0}^p)/x_{t_0}^p}{q_{x,t_0}(x_{t_0}^p; x_{t_0-1}^{(n-1)})} \cdot \prod_{t=t_{0,s}}^{t_0-1} e(y_t; x_t^{(n-1)}, s_{t_0}^p)$$

   Compute $W_{t_0}^p = w_{t_0}^p / \sum_{p=1:P} w_{t_0}^p$. In the numerator, the term $\pi_s(s_{t_0}^p)/x_{t_0}^p$ corresponds to $\pi_z(z_{t_0} \mid x_{t_0})$ where $z_t = x_t \cdot s_t$. The extra term $\prod_{t=t_{0,s}}^{t_0-1} e(y_t; x_t^{(n-1)}, s_{t_0}^p)$ accounts for the modified distribution of the observed read count between the last breakpoint on s, $t_{0,s}$, and $t_0 - 1$.

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

   (a) If $1/\sum_{p=1}^P (W_{t-1}^p)^2 > P/4$ then $a_t^p = p$ for $p = 1 : P$.

   (b) If $1/\sum_{p=1}^P (W_{t-1}^p)^2 \leq P/4$ then for $p = 1 : P-1$ draw index of ancestor particle a_t^p from weights $(w_{t-1}^p)_{p=1:P}$. Set $a_t^P = P$ and $(W_t^p)_{p=1:P} = 1/N$.

   (c) For $p = 1 : P-1$, draw x_t^p from $q_{x,t}(x_t^p; x_{t-1}^{a_t^p})$. Set $x_t^P = x_t^{(n-1)}$.

   (d) If $bk_{s,t}$ = 'no change' then for $p = 1 : P$, set $s_t^p = s_{t-1}^{a_t^p}$ and compute particle weights $w_t^p = W_{t-1}^{a_t^p} \cdot \frac{k_x(x_t^p; x_{t-1}^{a_t^p}) \cdot e(y_t; z_t^p)}{q_{x,t}(x_t^p; x_{t-1}^{a_t^p})}$; $W_t^p = w_t^p / \sum_{p=1:P} w_t^p$.

   (e) If $bk_{s,t}$ = 'change' then, for $p = 1 : P$, set $s_t^p = z_t^{(n-1)}/x_t^p$ and compute weights $w_t^p = W_{t-1}^{a_t^p} \cdot \frac{k_x(x_t^p; x_{t-1}^{a_t^p}) \cdot e(y_t; z_t^p) \cdot \mathbf{1}_{\{x_t^p>0\}} \cdot \pi_s(s_t^p)/x_t^p}{q_{x,t}(x_t^p; x_{t-1}^{a_t^p})}$; $W_t^p = w_t^p / \sum_{p=1:P} w_t^p$.
   The extra term in the numerator $\pi_s(s_t^p)/x_t^p$ corresponds to $\pi_z(z_t^p \mid x_t^p)$.

3. For $t = t_1$,

   (a) Update weights $w_{t_1}^p = W_{t_1}^p \cdot k_x(x_{t_1+1}; x_{t_1}^p) \cdot \prod_{t=t_1+1}^{t_{1,s}} e(y_t; x_t, s_{t_1}^p)$,where the last term corresponds to the distribution of counts up to the next breakpoint on s.

   (b) Backward sample $\{x, s\}_{t_0:t_1}^{(n)}$ from the filtering sample $(\{x, s\}_{t_0:t_1}^p)_{p=1:P}$ with filtering weights $(w_{t_0:t_1}^p)_{p=1:P}$ (algorithm 12).

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

4.2.4 Empirical Analysis of Exactness for CSMC and SIR

We design an algorithm for the detection of eventual errors in the implementation of our algorithm (due to implementation or method issues) and use it in the same time to perform empirical analysis of SMC accuracy. Its strategy relies on the idea of running an extended version of MCMC in which we add a step to each sweep where the observations $\mathbf{y}$ are sampled from their conditional distribution $y \mid x, \Theta$. The algorithm samples then the joint distribution y, x, Θ instead of $x, \Theta \mid y$. If the algorithm is precise, the empirical distribution of Θ should correspond exactly to the prior distribution (which is the marginal of the joint distribution y, x, Θ).

Our final algorithm relies on Conditional SMC updates of the hidden trajectories $\mathbf{x}$ and $\mathbf{s}$ which provide an exact MCMC algorithm (termed Particle Gibbs) for sampling their distribution given the observed $\mathbf{y}$. 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. For a finite number of particles, this simpler algorithm provides a trajectory that is only approximatively distributed according to the target conditional distribution. To illustrate the impact of using an approximate algorithm (MCMC based on SIR) instead of an exact algorithm (Particle Gibbs based on Conditional SMC) we run the extended algorithm using a toy model. This is a simplification of the Parseq model with a mixture kernel built from 'no change' and exponential moves $\pi(x_t \mid x_{t-1}) = \alpha \cdot \delta_{\{x_{t-1}\}}(x_t) + (1-\alpha) \cdot \lambda e^{-\lambda \cdot x_t}$. We use a Poisson distribution: $\pi(y_t \mid x_t) = \mathcal{P}(y_t, x_t)$ for the emission. In practice we fix $\lambda = 1$ and use for α a Beta prior of shapes 10 and 2 (giving a prior expectation of 10/12). In order to increase the convergence speed, the extended algorithm was run on a short sequence (T=200 bp). The number of particles and other SMC settings are detailed in table 4.1. We perform our analysis on SIR and CSMC filtering approximations. We study the effects of introducing backward sampling, of using 'good' and 'bad' proposals and of various SMC set-ups (see table 4.1). The results shown in figure 4.2 are illustrating the difference in precision between CSMC and SIR. We analyse the estimation of the parameter α, the kernel mixture weight. The histograms represent the posterior distribution of α obtained as a marginal approximation from $\pi(y_{1:T}, x_{1:T}, \alpha)$. In case of exact estimation, the red line representing the prior $\pi(\alpha) = \frac{\alpha^9 \cdot (1-\alpha)}{\beta(10,2)}$ should fit the empirical distribution (histogram).

Experiment	SMC	Backward sampling	Proposal	Number of particles (P)
A	SIR	No	q^k	200
B and B^*	SIR	No	q^*	200 (B) and 2000 (B^*)
C	CSMC	No	q^k	200
D and D^*	CSMC	No	q^*	200 (D) and 100 (D^*)
E	SIR	Yes	q^*	200
F and F^*	CSMC	Yes	q^*	200 (F) and 100 (F^*)

Table 4.1: List of SMC algorithms and corresponding set-ups submitted to accuracy analysis. We test two proposals: the prior kernel $q_t^k(x_t; x_{t-1}) = \pi(x_t \mid x_{t-1})$ and a mixture with a component that does not fit the target $q_t^*(x_t; x_{t-1}) = 0.95 \cdot \delta_{\{x_{t-1}\}}(x_t) + 0.05 \cdot \lambda e^{-2 \cdot x_t}$. We indicate in the last column the numer of particles used for the SMC approximation.

First, for the kernel prior proposal and P=200 particles the filtering methods SIR and the CSMC have comparable good performance (panels A and C).

Second, we decrease the fit of the proposal to the target and use a fixed weight mixture with an $\lambda = 2$ exponential parameter (model kernel has $\lambda = 1$). This proposal writes $q_t(x_t; x_{t-1}) = 0.95 \cdot \delta_{\{x_{t-1}\}}(x_t) + 0.05 \cdot \lambda e^{-2 \cdot x_t}$. For this proposal and P=200, the SIR filtering results in a significant decrease in accuracy. As expected, increasing the number of particles to P=2000 improves SIR filtering estimation (panel B^*). CSMC filtering requires less particles to provide exact estimation (panel D). While for P=200 results show a good accuracy, for P=100 the histogram does not fit the prior distribution. One important reason for the CSMC degeneracy is the low update length. Indeed going backward we update at iteration n on average 20 positions from of $x_{:T}^{(n-1)}$ for P=100 and 50 positions for P=200. Using the prior kernel proposal we updated 80 positions for P=200. A low update length implies a high inertia within PG sweeps.

Third, we use backward sampling (12) to reconstruct the trajectories. While the SIR algorithm has no significant increase in precision (panel E) we notice that CSMC results in good estimations for both P=200 and P=100 (panels F and F*).

In conclusion, for proposals that require a significant sample size to approximate the target and a moderate number of particles the SIR algorithm has markedly biased results. For the same set-up the CSMC leads to an accurate estimation.

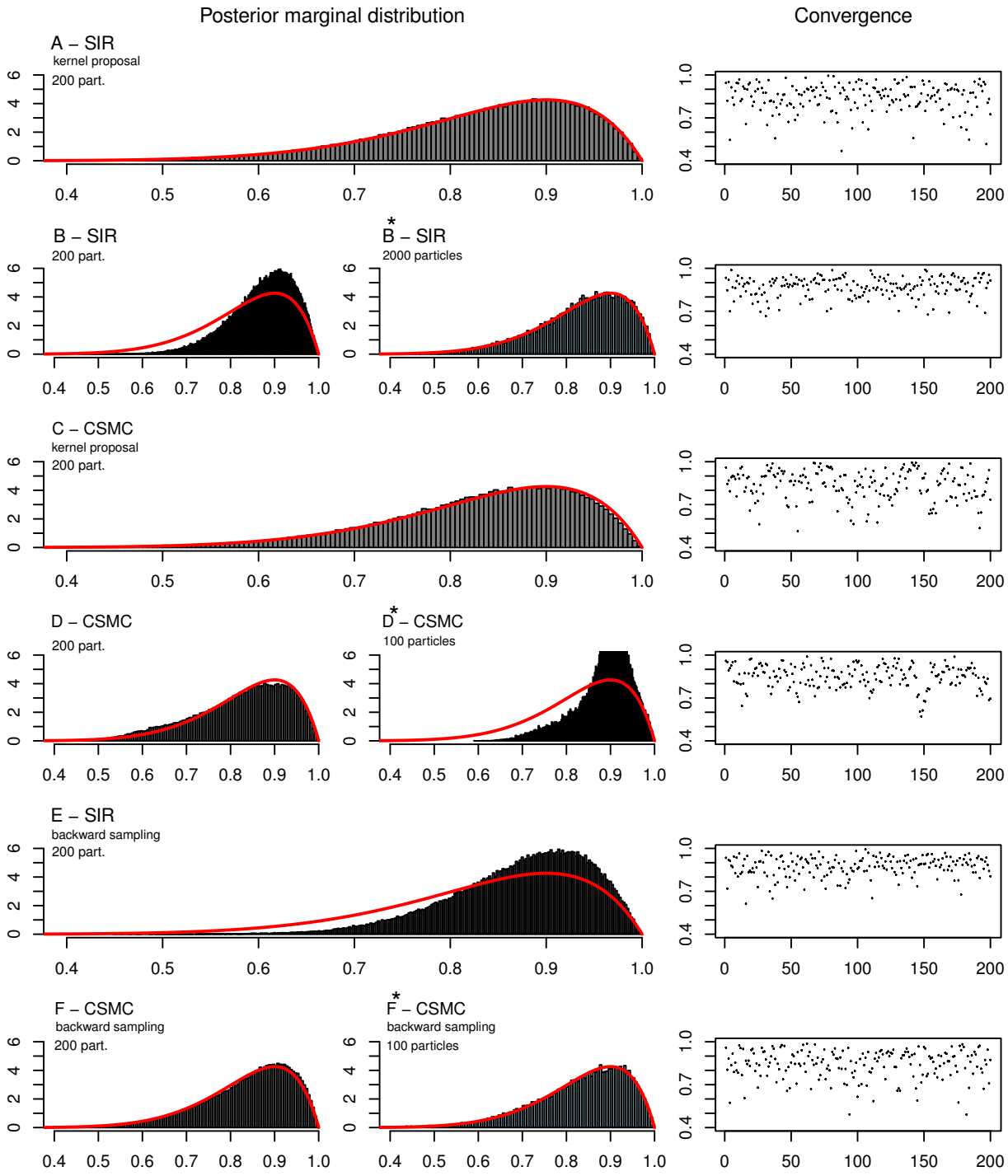

Figure 4.2: Analysis of exactness for SMC algorithms - a toy model. We show the estimation of the transition kernel parameter α. In the left we show histograms of the marginal distribution of α (from $\pi(y_{1:T}, x_{1:T}, \alpha)$). The red line represents the prior density of α. Exact estimation supposes that the marginal distribution corresponds to the prior. In the right we show the values of α along the 200k sweeps (thinning step of 10). For each scenario we indicate the SMC method, the number of particles, and we mentioned if we used the prior kernel proposal and if we performed backward sampling. We make a summary of the scenarios A, B, B*, C, D, D*,E, F, F* in the table 4.1.

4.2.5 The Choice of the Proposal Function

The efficiency of a SMC algorithm, here the Particle Gibbs, depends heavily on the choice of an appropriate proposal kernel. In this work we relied on two position-specific proposals $q_{x,t}$ and $q_{s,t}$ that aim at drawing new values of x_t and s_t near their target posterior distributions.

In the SMC context sampling needs to be done from the target $\pi(x_t \mid x_{t-1}, y_t)$. It is difficult to evaluate how the efficiency on the smoothed approximations of complete trajectories is determined by the efficiency of proposals that aim at filtering extensions. Therefore we do not limit to the analysis of instrumental functions approximating the optimal filtering proposal, $q_t(x_t; x_{t-1}) = \pi(x_t \mid x_{t-1}, y_t)$, and we investigate other classes of instrumental functions that try to approximate $\pi(x_t \mid x_{t-1}, y_{1:T})$. We review in section 2.4 the general validity criteria that a proposal function needs to fulfil. Besides these criteria a proposal needs to be efficient, i.e. needs to permit target approximation with a practical sample size and needs to be easy to sample from.

Simple proposal functions

First, as computational time is a significant issue, we ask ourselves if simple proposals like the prior kernel can be used for practical SMC set-ups.

The prior kernel is a proposal that is computational efficient and simplifies the particle weights computation to evaluating the emission (see section 2.4). Also, in our SSM model sampling from the transition kernel $\pi(x_t \mid x_{t-1})$ can be done directly. However, this proposal can not sample well the regions where data proves to be significantly different from the levels accounted by the transition kernel.

Then, we ask if we can use a proposal built upon the posterior of the emission density $q_t(x_t) = \pi(x_t \mid y_t)$. The target distribution is a mixture where components include Dirac mass functions ($\delta_{\{0\}}$ and $\delta_{\{x_{t-1}\}}$). Therefore the proposal is not valid (it exists $x_t = x_{t-1}$ such as $q_t(x_t) = 0$ and $\pi(x_t) > 0$).

Mixture proposal functions

We decide to design a mixture proposal that includes partially the transition kernel components and accounts at the same time for the observations. As observations are highly variable, we decide to account for a window k of observations starting from position t.

This proposal for the expression level x_t writes as

$$\begin{aligned} q_{xt}(x_t; x_{t-1}) &= \mathbf{1}_{\{x_{t-1}=0\}} \cdot \left[(1-\eta_q)\cdot\delta_0(x_t) + \eta_q \cdot \zeta_q e^{-x_t\zeta_q}\right. \\ &\quad \left. + \mathbf{1}_{\{x_{t-1}>0\}} \cdot \left[\alpha_{q1}\cdot\delta_{x_{t-1}}(x_t) + (1-\alpha_{q1})\cdot l_t(x_t)\right]\right., \end{aligned} \tag{4.2}$$

where $1-\eta_q$ and α_{q1} play the same role as the kernel parameters, i.e. represent the probability of not changing the expression level if the previous position has the value 0 and respectively above 0. The density $l_t(x_t)$ is itself a mixture

$$l_t(x_t) = \alpha_{q2}\cdot\delta_0(x_t) + \alpha_{q3}\cdot\mathcal{E}(x_t,\zeta_q) + \alpha_{q4}\cdot q_{y_{t:t+k-1}}(x_t)\,, \tag{4.3}$$

where $q_{y_{t:t+k-1}}(x_t)$ is a density that depends on the observations in the window $t : t+k-1$ (built as a posterior for a simplified model).

Initially we used an approximate emission model, i.e. a Poisson distribution of rate x_t. The Poisson has a Gamma density for its conjugate prior. If we consider an exponential prior distribution with rate ζ_p for the expression level and assume constant expression level, the posterior $\pi(x_t \mid y_{t:t+k-1})$ (a Gamma density) writes $\Gamma(x_t, \frac{\sum_{i=0:k-1} y_{t+i}+1}{k+\zeta_q}, \frac{1}{\sum_{i=0:k-1} y_{t+i}+1})$ of scale $k+\zeta_p$ and shape $\sum_{i=0:k-1} y_{t+i}+1$ (density parametrisation in 3.0.2).

Slow convergence in regions with high local scaling bias motivated a more complex approximation of $q_{y_{t:t+k-1}}(x_t)$. We approximate it as a Poisson of rate $x_t \cdot s_t$. In this case, the posterior $\pi(x_t \mid y_{t:t+k-1}, s_{t:t+k-1})$ writes as $\Gamma(x_t, \frac{\sum_{i=0:k-1} y_{t+i}+1}{\sum_{i=0:k-1} s_{t+i}+\zeta_q}, \frac{1}{\sum_{i=0:k-1} y_{t+i}+1})$ a Gamma distribution which, compared to the previous Gamma has scale $\sum_{i=0:k-1} s_{t+i}+\zeta_q$ in order to account for position scaling of the expression level. The density from 4.3 writes

$$l_t(x_t) = \alpha_{q2}\cdot\delta_0(x_t) + \alpha_{q3}\cdot\mathcal{E}(x_t,\zeta_q) + \alpha_{q4}\cdot\Gamma(x_t, \frac{\sum_{i=0:k-1} y_{t+i}+1}{\sum_{i=0:k-1} s_{t+i}+\zeta_p}, \frac{1}{\sum_{i=0:k-1} y_{t+i}+1}) \tag{4.4}$$

In practice, for the datasets presented in results, we have set $\zeta_q = 0.1$, $k = 3$, $1-\eta_q = 0.9$, $\alpha_{q1} = 0.9$ and $\alpha_{q2} = \alpha_{q3} = \alpha_{q4} = 1/3$.

For the local scaling s we build a $q_{s,t}$ following the same mixture principles:

$$q_{s,t}(s_t; s_{t-1}) = \frac{9}{10}\delta_{s^i_{t-1}}(s^i_t) + \frac{1}{20}\Gamma(s_t, \frac{\sum_{i=0:k-1} y_{t+i}+1}{k\cdot x_t+1}, \frac{1}{\sum_{i=0:k-1} y_{t+i}+1}) + \frac{1}{20}\mathcal{E}(s_{t+1}; rate=1).$$

Adaptive mixture proposal

In later versions of Parseq we try to optimize the weights of proposal components. For this we implement in the algorithm 14, the MPMC adaptive scheme (algorithm 10).

We design a mixture proposal with 6 components of weights $\alpha_q = (\alpha_{q0} : \alpha_{q5})$. Specifically, we enrich the mixture from 4.4 by having two components dependent on the observations $q_t(x_t; y_{t:t+2})$ with weight α_{q4} and $q_t(x_t; y_{t:t+8})$ with weight α_{q5} (k=3 and k=9 for the Gamma distribution from equation 4.4). The proposal for x_t writes

$$\begin{aligned} q_t(x_t; x_{t-1}, \alpha_q) = \quad & \mathbf{1}_{\{x_{t-1}=0\}} \cdot \quad \big[\alpha_{q0} \cdot \delta_0(x_t) + (1-\alpha_{q0}) \cdot \mathcal{E}(x_t; \zeta_q)\big] + \\ & \mathbf{1}_{\{x_{t-1}>0\}} \cdot \quad \big[\alpha_{q1} \cdot \delta_{x_{t-1}}(x_t) + \alpha_{q2} \cdot \delta_0(x_t) + \alpha_{q3} \cdot \mathcal{E}(x_t, \zeta_q) + \\ & \qquad \alpha_{q4} \cdot q_{y_{t:t+2}}(x_t) + \alpha_{q5} \cdot q_{y_{t:t+8}}(x_t)\big]. \end{aligned} \tag{4.5}$$

Algorithm 14 Particle Gibbs with sequential adaptive proposal

Objective: optimize the weights $\alpha_q = (\alpha_{qd})_{d=0:5}$ of a mixture proposal $q_t(x_t; x_{t-1}, \alpha_q)$ defined in equation 4.5 in order to minimize its divergence from the target $\pi_t(x_t \mid x_{t-1}, y_{1:T})$.

For $n = 1 : N$:

1. run CSMC aiming at $x_{1:T} \mid \Theta^{(n-1)}, s_{1:T}^{(n-1)}, y_{1:T}$ using $q_t(x_t; x_{t-1}, \alpha_q^{(n-1)})$ proposal (algorithm 11);
2. draw $x_{1:T}^{(n)}$ using backward sampling (algorithm 12);
3. for each position t compute posterior proposal weights:
   - for $d = 1 : 5$ compute $\alpha_{td}^{(n)} \propto \alpha_{qd}^{(n-1)} \cdot \frac{q_{td}(x_t^{(n)})}{q_t(x_t^{(n)})}$ such as $\sum_{d=1:5} \alpha_{td}^{(n)} = 1$,
   - for $d = 0$ compute $\alpha_{t0}^{(n)} = \alpha_{q0}^{(n-1)} \cdot \frac{q_{t0}(x_t^{(n)})}{q_t(x_t^{(n)})}$;
4. for $d = 0 : 5$ compute average proposal weights $\alpha_{qd}^{(n)} = \sum_{t=1:T} \alpha_{td}^{(n)}$.

Continue with further non adaptive Particle Gibbs sweeps using a proposal with component weights $\alpha_q^{(N)}$.

We present in figure 4.3 results for using various proposals: the prior kernel, the mixture and the mixture with adapted weights. We illustrate this on for runs done with

Parseq (using Particle Gibbs with backward sampling) on *E. coli* real and synthetic data-sets (expression and counts generated according to the Parseq model). First, in-line with other results shown in figure 4.2, we observe that increasing the number of particles P greatly improves the update frequency for all proposal densities. Second, the adaptive proposal improves the update frequency for both average and low number of particles. For a low number of particles ($P = 20$) there is a remarkable difference (bottom-left plot). When the target function is well approximated (by using 'good' proposals or high number of particles) the difference is less important. Third, we notice that for high expression values, the proposals containing components dependent on observations have better update frequency. Interestingly, this difference can be noticed also for the synthetic data-set where the prior kernel proposal is equal to the simulation model kernel.

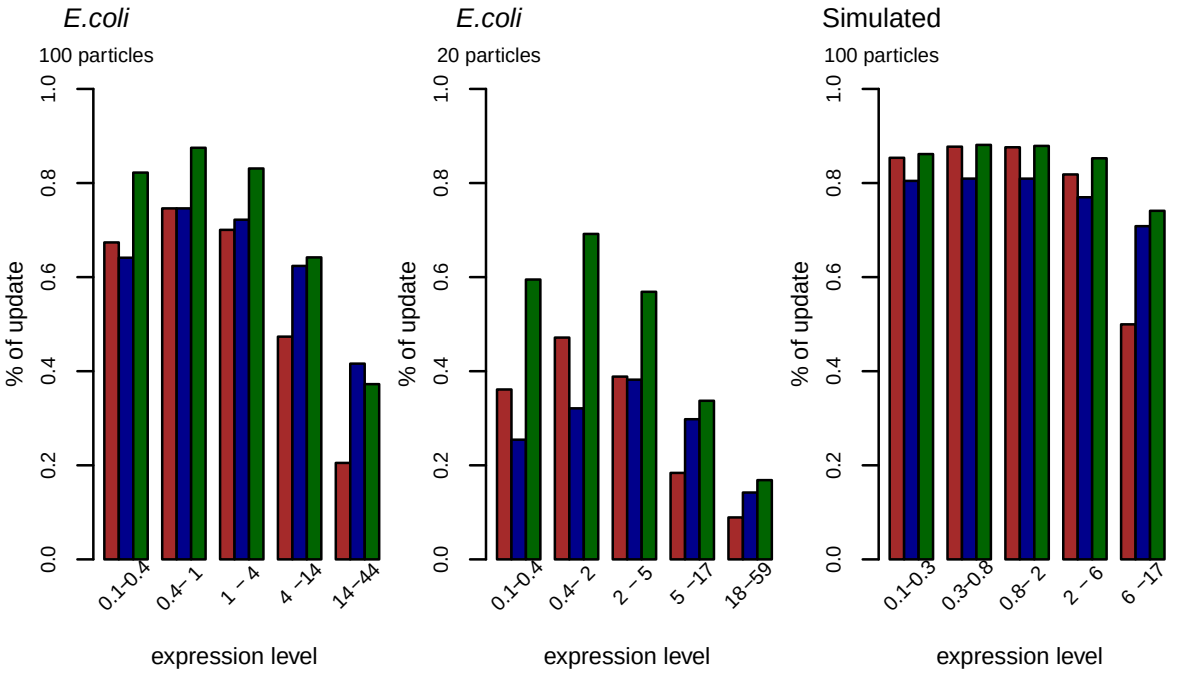

Figure 4.3: The performance of PG for different proposals for the approximation of the expression level. Data: 1 Mbp from *E. coli* real and synthetic data. SMC set-up: 220 sweeps, burn-in of 20, tinning step 5, number of particles P=100 or P=20. Brown: prior kernel $\alpha_q = (0.99, 0.97, 0.15, 0.15, 0, 0)$; Blue: mixture of kernel and observation components with fixed weights $\alpha_q = (0.99, 0.9, \frac{0.1}{4}, \frac{0.1}{4}, \frac{0.1}{4})$; Green: mixture of kernel and observation components with adapted weights $\alpha_q \approx (0.999, 0.99, 0.001, 0.004, 0.003, 0.006)$ for all runs. X -axis: expression level ranges $[e_i, e_{i+1})$. The frequency of particle update is computed as $\frac{1}{N}\sum_{n=1:N}\mathbf{1}_{\{x_t^{(n+1)} \neq x_t^{(n)}\}}$ for $\bar{x}_t \in [e_i, e_{i+1})$.

4.2.6 Estimation of the 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, a, κ_s and α_s) are estimated from the characteristics of the read-counts without hidden path reconstruction (section 3.3). We choose here to estimate these parameters beforehand in order to improve the MCMC mixing with respect to the other parameters.

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

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

We show in section 4.2.10 the posterior distribution of these parameters and convergence plots for the *S. cerevisiae* 1 data set.

4.2.7 Breakpoint Posterior Identification with Local Score

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 use for this purpose a local score approach (used in a slightly different form also by Nuel [2006]).

The 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, $z_t = \pi(b_{x,t} = \text{'shift'}, x_t > c \mid \mathbf{y}, \Theta)$ where c is the cut-off on expression level (see subsection 4.3.3). We select regions with positive score from the first positive value to the maximum local score in the region. To avoid overlooking downstream high scoring segments, after the end

position of each of these segments, the current position t is set to the one following the local maximum and the score s_t is set back to a null value. The penalty value m represents a correction for background signal. In the same time the value of m controls the minimal distance between consecutive breakpoints. It can be computed from the mean value over the whole sequence or locally. In practice, for the two datasets *S. cerevisiae* and *E. coli* m was set to 0.005 and c at respectively 0.1 and 0.5.

For each high scoring segment defined by the above procedure, we computed the cumulated probability of breakpoint $\sum_{t=t_1}^{t_2} \pi(bk_{x,t} = \text{'shift'}, x_t > c \mid \mathbf{y}, \Theta)$ between the segments end-points t_1 and t_2. We use 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.

4.2.8 Reconstruction of Transcription Units

While it is important for practical reasons, defining transcription units might hinder the rich information provided by the continuous transcription profile. By the standard definition, the transcription units should have homogeneous expression level and be delimited by breakpoints in the expression profile. In a larger sense, the transcribed regions can be defined as those regions that have continuous high transcription probability. With the Parseq results we can aim for the reconstruction of both the transcription units and transcribed regions.

First, we can reconstruct units with homogeneous expression level by considering regions contained between two breakpoints estimates. If the breakpoint uncertainty leads to missing start/end breakpoints we can limit the reconstructed units only within high transcription probability regions. It is also important to mention that, for breakpoints within a transcribed regions, the expression profile cannot be used to disentangle between the possible cases of adjacent or overlapping transcripts.

For the reconstruction of transcribed regions we can use the profile of transcription probability approximated during Parseq. Several ad hoc methods can be used to achieve this purpose. In one of them we delimit regions with continuous transcription probability

above a given threshold. We then merge the regions that are separated by a distance that has no biological meaning and discard regions that are smaller than the minimum expected transcript length. In another method, we use a strategy similar to the local score and construct the continuous regions where locally the average transcription probability is above a given threshold. We detail the last proposed strategy in the section 5.3.4 for the reconstruction of DE units and illustrate also transcript reconstruction in figure 6.2.

4.2.9 The Parseq Work-flow

By design the PG algorithm (section 4.2.1) permits to tackle parameter estimation and transcriptional landscape reconstruction simultaneously but our software Parseq subdivides the problem in three successive steps for practical reasons (figure 4.4).

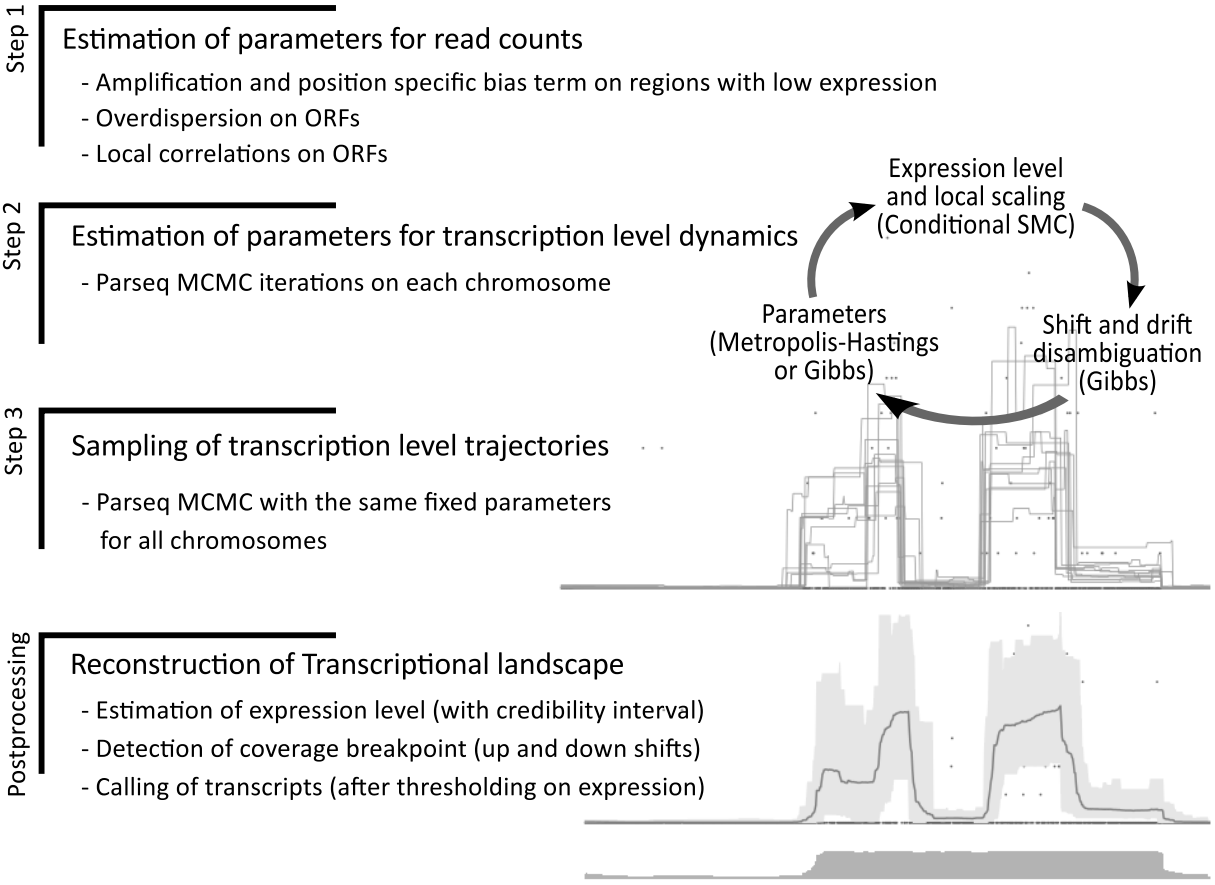

Figure 4.4: The Parseq work-flow: from parameter estimation to reconstruction of transcriptional landscape.

The parameters of the read-count emission model are estimated and the emission density corresponding to the different values of $x_t s_t$ are tabulated (step 1). PG iterations are too time-consuming to be performed on a single CPU for genomes of moderate sizes such as the yeast *Saccharomyces cerevisiae* ($\approx 12\,\mathrm{Mbp}$). The time and memory complexity are $O(P \cdot T \cdot N)$ and respectively $O(T)$ with T the sequence length, P the number of particles and N the number of PG sweeps. In order to distribute computation on independent CPUs, we decide to subdivide each chromosome in fragments ($\approx 1\,\mathrm{Mbp}$ each), to perform

parameter estimation separately on these fragments, and then to select a common set of parameters based of the obtained results (step 2). Posterior sampling of transcriptional landscape trajectories $\mathbf{x}$ is then carried out on a different CPU for each genome fragment, but with common parameters (step 3). With an Intel Core i7-3610QM CPU @ 2.30GHz, each complete sweep of the MCMC algorithm was recorded to take $\approx 1\,\text{min}$ for 1 Mb using $P = 150$ particles in each Conditional SMC updates. In the results shown in this chapter, we use 2200 sweeps, including 200 burn-in sweeps, for parameter estimation (step 2), and 2200 sweeps for making predictions at fixed parameters (step 3). For these results we do not incorporate the backward sampling step. On a 4 CPU computer the complete procedure took slightly less than 3 days for a 12 Mb genome with this algorithm set-up. The recent addition of the backward sampling and the proposal optimisation (discussed in section 2.4) made possible the utilisation of a smaller number of particles ($P = 50$). This greatly improved computation time and make it possible to obtain results for a 4Mb genome (like B.*subtilis*) overnight.

The output of the algorithm is a sample of transcriptional landscape trajectories drawn from $\mathbf{x}|\mathbf{y}, \Theta$ that conveys rich information about the actual transcriptional landscape. Here these trajectories served to estimate the expected value of x_t, the 95% credibility interval of x_t, and the probability of $x_t > 0$ (transcribed position), together with the probability of the different types of breakpoints along the sequence. Because of the posterior uncertainty on the exact position of each breakpoint we further aggregate the breakpoint probabilities at adjacent positions into small regions with high cumulative probabilities using a local-score approach (section 4.2.7). According to the direction of the change in expression level, the breakpoints were identified as up-shifts or down-shifts. In order to better distinguish genuinely expressed regions from (biological or technological) background noise we also realize the relevance of computing the probability for x_t to be above a selected cut-off and to predict the breakpoints that lead the trajectory $\mathbf{x}$ above this cut-off. Transcriptional landscape reconstruction is illustrated on fig. 4.5.

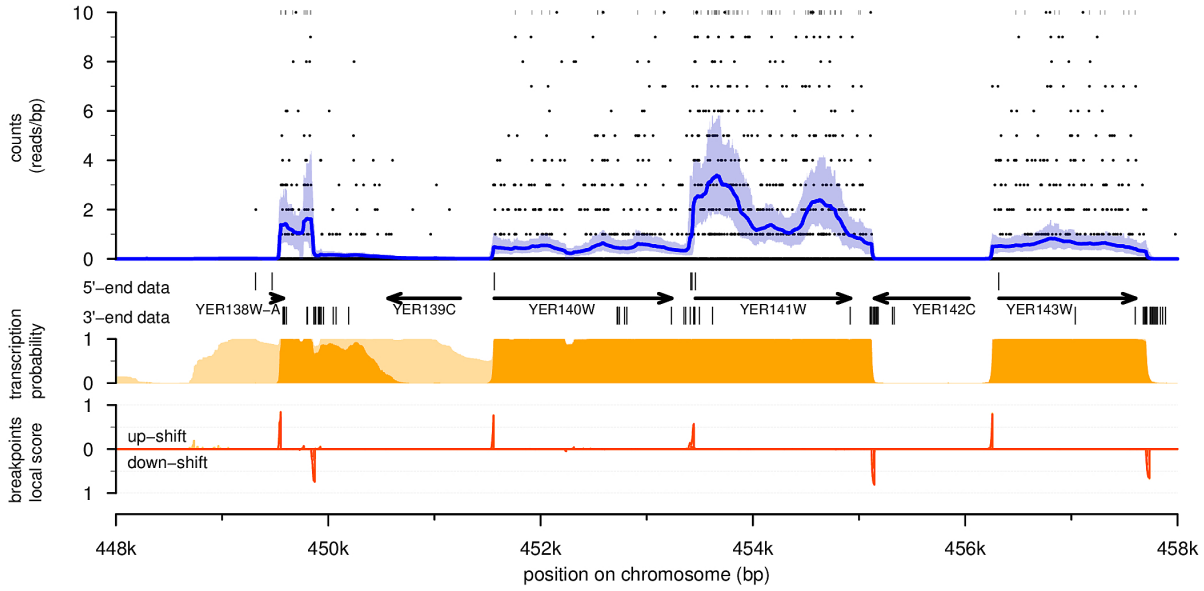

Figure 4.5: Transcriptional landscape reconstruction with Parseq. Example of results on a 10 kbp region of the first strand of *S. cerevisiae* chromosome V (dataset SRR121907). From top to bottom: read counts (dots) and the estimated expression profile (blue line) with its 95% credibility interval (light blue area); annotated CDSs (arrows) complemented with specific data sets of 5'-ends and 3'-ends (brown); probability of transcription with a cut-off on expression level set to 0^+ (light orange) or 0.1 reads/bp (orange); Local score in high scoring segments for the detection of breakpoints associated with up-shifts and down-shifts (red). This example illustrates the detection of overlapping transcription units (up-shifts before YER140W and YER141W) and incomplete termination sites (down-shift after YER138W-A).

4.2.10 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 behaviour 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 fix $\lambda_u = \lambda_d = 5.0$ which corresponds to a drift average change of 20%.

The table 4.2 summarizes the results obtained on *S. cerevisie* 1 and *E. coli* 2 datasets. The figure 4.6 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				
a	1.9	-	6.3	-
κ	1.2	-	0.6	-
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	2.9	-	4.6	-
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 4.2: Parseq parameters estimated on *S. cerevisie*1 and E. *coli* 2 data-sets. Parameter estimates are first obtained separately for each chromosome subdivision of $\approx 1Mb$ with Parseq. We average 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').

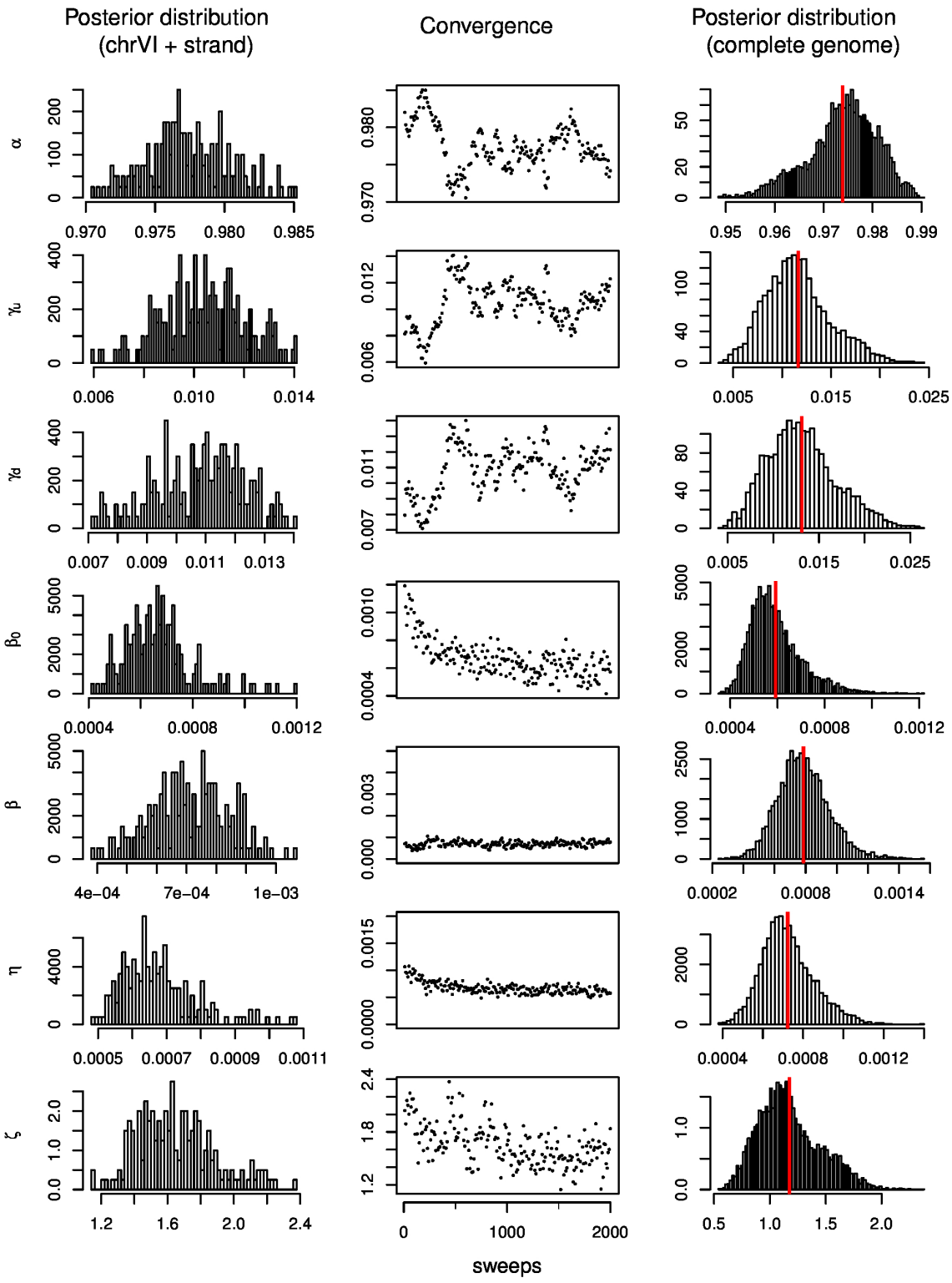

Figure 4.6: Parameter estimation on the *S. cerevisiae* 1 dataset: chromosome VI strand+ (sequence size 270kbp; left histogram and convergence plot) and complete genome (2.400kbp, right histogram). Left and right columns: histograms of the sampled values approximating the marginal posterior distributions. The complete genome histogram includes sampled values from for all chromosomes and both strands. Middle column: convergence plot along 2000 sweeps with a thinning step of 10 and excluding a 200 sweeps burn-in.

4.3 The Evaluation of Parseq Results

The accuracy of transcriptional landscape reconstruction was assessed from two different standpoints: the number of transcribed positions that can be correctly called based on the estimated value of x_t, and the number of transcript 5'-ends and 3'-ends at less than 50 bp of an identified up-shift and down-shift, respectively. To establish the lists of predictions we use a probability cut-off set to 0.5 for both the probability of $x_t > c$ (where c is an expression threshold) and the cumulative probability of shift in the small region delineated by local-score approach. When comparing the predictions with a reference annotation we needed to take into account that Parseq models the distribution of the 5'-end of the reads. For this reason, the regions predicted as transcribed by Parseq were extended of l_3 bp on their 3'-ends and the same correction needs to be applied to the predicted down-shifts before comparing with transcript 3'-ends (adjusted to 50 bp for the simulated data set). To report results in terms of sensitivity and positive predictive values (PPV) we compute the fraction of the true positives that could be matched to a prediction ($\frac{TP}{TP+FN}$) and the fraction of the predictions that could be matched to a true positive ($\frac{TP}{TP+FP}$).

Parseq predictions were systematically compared with the results of Cufflinks v2.1.1 [Trapnell et al., 2010], a method for transcript assembly which is based on read overlapping. For *E. coli* we compare our results also with Rockhopper [McClure et al., 2013]. Rockhopper makes use of existing annotation to estimate the transcription boundaries and read count distribution. To compare Cufflinks, Parseq and Rockhopper on the same basis we design a strategy to use Rockhopper to obtain transcript boundaries estimates without using annotation but in the same time to allow the estimation of read counts distribution. We divide the genome in 10 regions and we run Rockhopper 10 times when each time we exclude from the annotation one region. We assemble results obtained for each region where annotation is missing to obtain genome wide transcript boundaries estimates.

4.3.1 Evaluation of Results on Synthetic Data

The difficulty to find a reference annotation that could be considered as a gold standard motivated the idea of starting the results analysis with synthetic data sets. If for SMC performance we generate synthetic data using a model incorporating the read count model

described in section 3.2, for evaluating the results we use Flux simulator v1.2 [Griebel et al., 2012] (an application aiming at modelling RNA-Seq experiments *in silico* that tries to mimic the protocol steps). RNA-seq reads of length 50 bp were simulated specifying uniform RNA fragmentation. We allow 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 behaviour we increase the fragmentation rate up to an average fragment length of 20 (`FRAG_UR_DO 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).

For both real and synthetic data-sets we perform the alignment using Bowtie 1 v0.12.7 [Langmead et al., 2009] allowing only 1 mismatch in a 5 bp seed (`-n 1`), and discarding multiple alignments (`-m 1`). We use IGVTools [Thorvaldsdottir et al., 2013] to compute the 5' end read counts.

First we ask how estimation results change with increasing coverage. Strand-specific datasets of increasing sequencing depth (between 0.025 and 0.4 reads/bp after mapping) were simulated with the Flux simulator using the sequence and annotation of the *Saccharomyces cerevisiae* S288C chromosome IV. We increase the depth according to two scenarios: i) for scenario 1 we keep constant the initial number of mRNA molecules (30k) and vary amplification parameters (reads to molecules ratio from 5:1 to 80:1) and ii) for scenario 2 we maintain constant the amplification coefficient(reads to molecules ratio of 20:1) and vary the initial number of mRNA molecules from 7,5k to 120k. The sequencing depth increased similarly in both scenarios from 150k to 2400k reads.

The results obtained on synthetic data are summarized in fig. 4.7. While both Parseq and Cufflinks perform well when the depth of sequencing exceeds an average of 0.12 reads/bp, below this level differences between the two methods become evident. Even though they do not have the same sensitivity-specificity trade-off, it appears clearly that the results obtained by Parseq are better. The model-based approach adopted in Parseq makes it possible to extrapolate transcription across coverage gaps, and this results in a better calling of transcribed positions (not shown) and transcript borders. The mechanis-

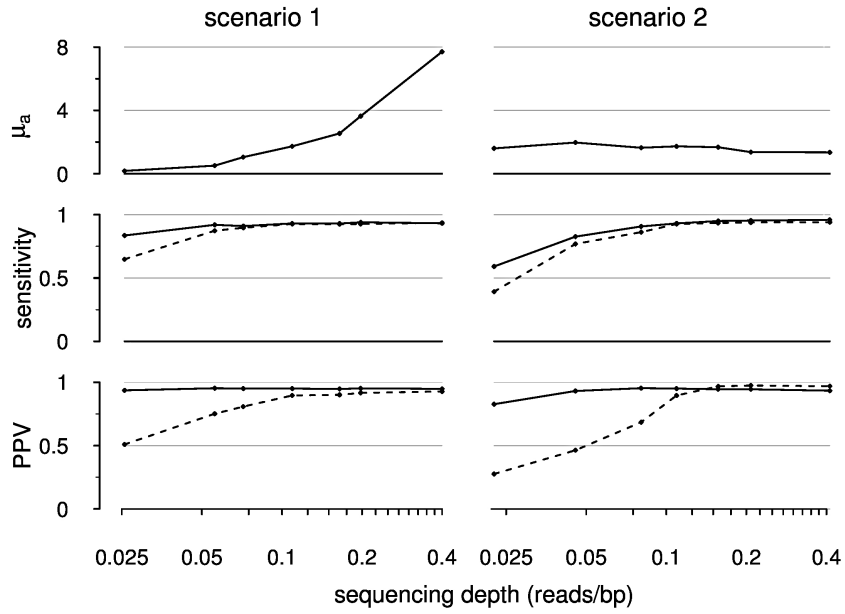

Figure 4.7: Impact of sequencing depth on transcript borders prediction in synthetic data. Two scenarios were considered to achieve higher sequencing depth: increasing the amount of amplification (left column) or increasing the number of initial molecules before amplification (right column). The evolution of the amplification coefficient μ_a estimated by Parseq distinguishes the two scenarios (top row). The results of Parseq and Cufflinks (default parameters) are represented by continuous and dashed lines, respectively (middle and bottom rows). The results were very similar for 5'-ends and 3'-ends and were pooled here.

tic interpretation of our new emission model is also well supported by the results: Parseq estimation of the amplification coefficient (μ_a, top plots) distinguishes remarkably well the two scenarios considered in our simulations where sequencing depth increases either as a consequence of higher amplification or as a consequence of higher number of initial molecules sampled.

4.3.2 Evaluation on Real Data

On synthetic data both the model-based approach of Parseq and the read-overlapping approach of Cufflinks perform well at detecting transcribed positions and transcript borders once the sequencing depth becomes high enough (0.12 reads/bp in our simulations). However, despite the efforts made on the simulation pipeline to mimic the different types of artifacts, the synthetic data does not have the complexity of a real data set.

For evaluation on real data we chose strand-specific, single-end, data sets from two major model micro-organisms: the yeast *Saccharomyces cerevisiae* (dataset *S. cerevisiae* 1) and the bacterium *Escherichia coli* (dataset *E. coli* 2). The *S. cerevisiae* 1 data-set was sequenced on a SOLiD platform and published in a study on regulatory non-coding RNAs [Dijk et al., 2011]. It has a read-length of 50 bp and a sequencing depth of 1.6 reads/bp after mapping. The *E. coli* 2 data-set was sequenced on an Illumina platform and published toghether with the presentation of the Rockhopper work-flow for bacterial RNA-Seq data processing [McClure et al., 2013]. It has a read-length of 100 bp and a sequencing depth of 2.4 reads/bp after mapping.

As a reference annotation for the transcribed positions in *S. cerevisiae*, we relied on the 5874 coding sequences (CDSs) found in the *S. cerevisiae* database SGD [Cherry et al., 2012] and lists of untranslated regions (UTRs) mapped from RNA-Seq experiments in Yassour *et al.* (2009) (5200 5'UTRs and 5295 3'UTRs). To better assess the accuracy of the prediction of transcripts 5'- and 3'-ends, we also include comparison with experimental data that aimed at mapping precisely these sites: 4393 transcriptional start sites (TSSs) [Zhang, 2005], and 7977 polyadenylation sites (pAs) [Ozsolak et al., 2010]. For *E. coli* we use annotations available in the RegulonDB database [Salgado et al., 2013] (2438 promoters and 2647 operons) and also the sequence-based predictions of 2260 rho-independent transcription terminators obtained with Petrin software [Carafa et al., 1990].

Importance of drift and local correlations. On real data, taking into account the local correlations and the drift prove to be important as indicated not only by their estimated values but also by the accuracy of transcript border detection. IBy monitoring the accuracy in terms 5'-ends and 3'-ends detection, we assessed the effect of these two model components on the quality of the inference. The results are reported in table 4.3 and confirm that taken individually the drift and the local correlations improve the results. Moreover, the results also demonstrate that the two terms are complementary rather than redundant since their combination lead to further improvements. As reported in table 4.3, the introduction of local scaling allowed a dramatic decrease of the coefficient of variation of the estimated expression level within annotated genes. At this point, we would like to recall that this better smoothing of the expression level comes with of a huge increase of the credibility intervals (4.5). That illustrates the cost implied by the

Parseq components	included in the model			
drift[a]	+	+	-	-
autocorrelation[b]	+	-	+	-
5'-ends number	6,689	13,881	15,994	31,428
TSS sensitivity	64%	70%	74%	79%
TSS PPV	48%	28%	25%	15%
3'-ends number	6,287	11,880	16,613	32,357
pAs sensitivity	60%	63%	70%	74%
pAs & 3'UTR PPV	57%	34%	29%	17%
CV[c] within CDSs	0.37	0.57	0.43	0.59

Table 4.3: Impact of drift and local scaling. Results obtained on *S. cerevisiae* 1 chr. IV (both strands) with expression cut-off 0.1 reads/bp. [a] drift is removed by setting $\gamma_u = \gamma_d = 0$. [b] short-range autocorrelation is removed by setting $\alpha_s = 0$.[c] coefficient of variation. PPV represents the Positive Predictive Value.

existence of correlated overdispersion and the difficulty that goes with it when comparing expression levels between regions of the genome.

The evaluation of the complete Parseq model. Table 4.4 presents a detailed breakdown of the results according to the different sets of reference annotations which could be considered to assess accuracy. In this context, we find that the probability of $x_t > 0$ (expression cut-off 0^+) is not necessarily the most relevant to compare the prediction of transcribed positions with a reference annotation. The best trade-offs are obtained near 0.1 reads/bp on the *S. cerevisiae* data-set, and 0.25 reads/bp on the *E. coli* data-set. These values are in agreement with the presence of a large number of positions associated with low expression level, resembling a background noise (section 4.3.3). The accuracy of the detection of transcribed position is remarkable (e.g., 83% sensitivity, 90% PPV with the 0.1 reads/bp expression cut-off on *S. cerevisiae*) but very similar to Cufflinks. In keeping with our observations on synthetic data, this suggests that detecting transcribed positions is easy at high sequencing depth and consequently the model-based approach implemented in Parseq provides only small benefits. The accurate identification of transcript borders is by far more challenging. For instance, on *S. cerevisiae* 5'-ends, with

the same 0.1 reads/bp expression cut-off, the sensitivity reaches 64% and the PPV 48%. On *E. coli*, PPVs remain acceptable but sensitivity values are much lower. This could be due to a combination of: lower quality of the data (μ_a estimated to 6.15 in *E. coli* versus 1.18 in *S. cerevisiae*, adjusted l_3 is 50 bp for *S. cerevisiae* versus 160 bp for *E. coli*); lower quality of the annotation taken as reference (e.g., Petrin predictions are expected to contain substantial numbers of false positives and false negatives); higher proportion of genes with low or no expression and thus for which promoters and terminators cannot be detected (with the 0^+ expression cut-off, sensitivity for detection of transcribed regions is only 0.81 in *E. coli* versus 0.91 in *S. cerevisiae*). On both data sets and for 5'-ends and 3'-ends alike, Parseq results are consistently better than the ones obtained by Cufflinks, particularly in terms of sensitivity. This confirms our expectations as Cufflinks reconstruction ignores the possibility of overlapping transcripts and thus overlooks transcript-ends in these configurations. We also include in our comparison the predictions made on *E. coli* by Rockhopper (Table 4.4). As we are interested here in de-novo predictions but this software could not run without annotations, we discard successively the annotation on one-tenth of the genome and recorded the predictions on it. Parseq and Cufflinks provide results markedly better than Rockhopper in this comparison set-up.

		S. cerevisiae			
		Reference	Parseq	Cufflinks	Rockhopper
Transcripts	Sensitivity	CDSs & UTRs	0.83 (0.91)	0.83(0.87)	–
	PPV	CDSs & UTRs	0.90 (0.68)	0.90 (0.81)	–
5' End	Number		6,689 (8,353)	5,484 (13,622)	–
	Sensitivity	TSSs	0.64 (0.65)	0.43 (0.45)	–
	PPV	TSSs & 5'UTRs	0.48 (0.4)	0.49 (0.22)	–
3' End	Number		6,287 (7,440)	5,484 (13,622)	–
	Sensitivity	pAs	0.60 (0.62)	0.43 (0.44)	–
	PPV	pAs & 3'UTRs	0.57 (0.51)	0.51 (0.22)	–
		E. coli			
Transcripts	Sensitivity	Operons	0.56 (0.81)	0.60 (0.75)	0.21 (0.39)
	PPV	Operons	0.76 (0.57)	0.72 (0.61)	0.91 (0.86)
5' End	Number		1,846 (2,193)	1,577 (7,962)	2,949 (4,401)
	Sensitivity	Promoters	0.24 (0.25)	0.15 (0.23)	0.12 (0.19)
	PPV	Prom. & 5'Oper.	0.49 (0.42)	0.34 (0.11)	0.24 (0.23)
3' End	Number		1,327 (1,342)	1,577 (7,962)	2,949 (4,401)
	Sensitivity	Terminators	0.12 (0.11)	0.08 (0.13)	0.03 (0.08)
	PPV	Term. & 3'Oper.	0.35 (0.32)	0.24 (0.08)	0.07 (0.11)

Table 4.4: Detection of transcribed positions and transcript borders on data-sets *S. cerevisiae* 1 and *E. coli* 2. Predictions and reference data were matched based on a ± 50 bp distance cut-off. Outside parentheses: results obtained after applying a stricter 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/transcript for Cufflinks, $z = 0.2$ for Rockhopper. Between parentheses: 0^+ reads/bp for Parseq, 5 fragments/transcript for Cufflinks, $z = 0.01$ for Rockhopper.

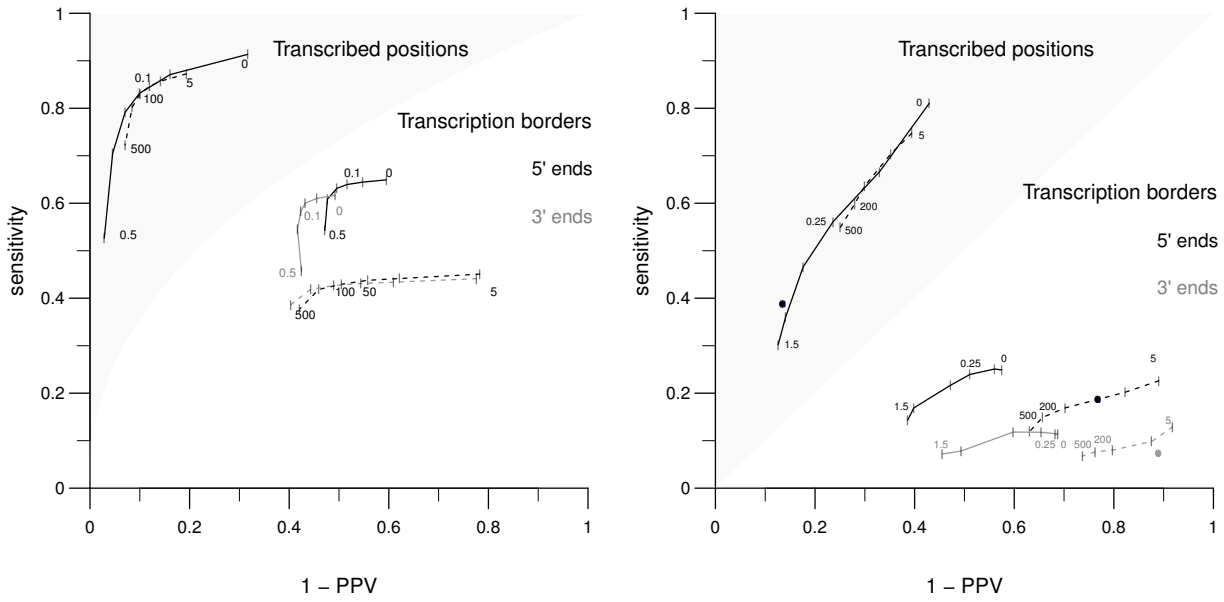

Figure 4.8: Impact of varying the expression cut-off on the accuracy of predictions. Results are shown for the datasets *S. cerevisiae* 1 and *E. coli* 2 (right panel). The performance obtained for three types of features are reported on the same plot: transcribed positions (grey area, upper right corner), transcript 5'-ends (black lines) and 3'-ends (grey lines). Solid lines: 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*. Dashed lines: results of Cufflinks for minimum fragments required per transcripts increasing from 5 to 500. Bullet points: *E. coli* results of Rockhopper with $z = 0.2$.

Completing results presented in table 4.4 we show in figure 4.8 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. It also 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 SI from [Mirauta et al., 2014]).

4.3.3 Choice of a Cut-off on Expression Level

We notice in our estimates of $\mathbf{x}$ 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 (biological or experimental) 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 examine the marginal distribution of the estimated expression levels $\mathbf{x}$ (figure 4.9). For both data sets we observe an accumulation of position associated with low expression inconsistent with the behaviour of an exponential distribution (which is linear in log density). In practice, we set our expression cut-off to 0.1 reads/bp for *S. cerevisiae* 1 and 0.25 reads/bp for *E. coli* 2 when calling transcripts and breakpoints.

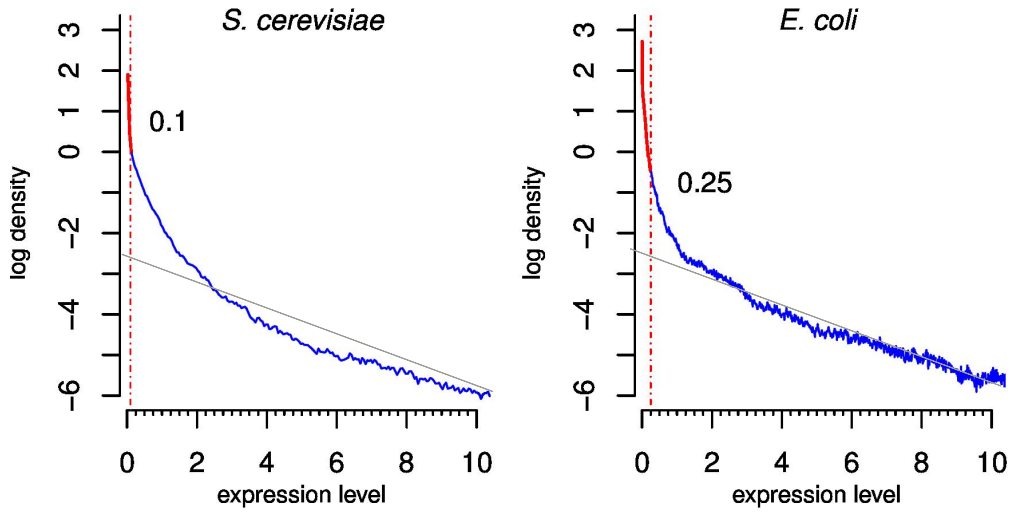

Figure 4.9: Marginal distribution of the estimated expression levels $\mathbf{x}$ on *S. cerevisiae* 1and *E. coli* 2 data sets. In this log-density *vs.* expression level plot an exponential distribution corresponds to a straight-line, as observed in the left part of the blue segments. Below 0.1 reads/bp in *S. cerevisiae* 1 and 0.25 reads/bp in *E. coli* 2 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).

Chapter 5

The analysis of multiple conditions

We present in this chapter results for the detection of positions and regions exhibiting differential expression (DE) between two conditions. Our approach approximates first the relative change in expression level for each position, then estimates the DE at a base-pair resolution and finally reconstructs DE regions.

Starting from two or more controlled experiments two types of question can be addressed:

(a) which genomic regions exhibit differential expression (DE) between conditions,

(b) how may the transcript annotation be improved when combining multiple datasets.

In this section we focus on the DE analysis. In statistical terms, the DE question translates into detecting significant changes of expression level, after accounting for possible sources of experimental and biological variability. While the classic approaches to identifying DE use predefined transcription units, e.g. genes and exons, we address this question in the context where such units are not available. This problem setting allows the identification of DE units that might not correspond to existing annotation. In the same time, we are confronted with a new problem, which is to reconstruct the DE units from a genome wide comparison of read counts coverage.

We want to provide a statistically sounded way of estimating the fold change and calling regions exhibiting DE above a given fold change level c between two data-sets. We base our method on the separate estimation of the expression level on each data-set, denoted $\mathbf{u} = (u_t)_{t\geq 1}$ and $\mathbf{v} = (v_t)_{t\geq 1}$, with Parseq (section 4.2.9). We present first

simple methods to compute genome-wide profiles of differential expression, i.e. obtain fold change estimates at base-pair resolution. Using Parseq expression profile samples we can compute point estimates of the fold change and also provide a credibility interval. The next step consists in calling DE at position level. For this we consider a DE threshold and compute the DE probability, which is the probability of the fold change to be above the DE threshold. Finally, we reconstruct the DE units, i.e. continuous regions where locally the DE probability is above a given threshold.

In the conclusion chapter we also discuss the design of a SSM model built for several sets of observations (conditions). This aims on one hand to estimate breakpoints in the expression profile that do not present differences between conditions (consensus breakpoints) and on the other hand to estimate directly the DE profile.

5.1 The Fold Change at Base-Pair Resolution

Our goal is to estimate the fold change at a base-pair resolution, that is to estimate the change between u_t and v_t where t is the position on the genome. For this reason, all our variables refer to the position resolution and we often omit to write the position index t.

A direct point estimate is provided by the ratio of posterior means $\hat{r}_{\mathrm{RM}} = \frac{\bar{u}}{\bar{v}}$ where $\bar{u}$ and $\bar{v}$ are expectations of the posterior distributions of the expression levels u and v as sampled by the Parseq algorithm. To incorporate the information on uncertainty embedded in the posterior distribution we also consider fold-change estimate based on the posterior distribution of the ratio $r = \frac{u}{v}$. A natural way of doing it is to consider the empirical distribution of the sample $(r)_{1\leq i\leq N^2} = (\frac{u^{i_u}}{v^{i_v}})_{1\leq i_u\leq N, 1\leq i_v\leq N}$ where N is the sample size drawn from each posterior using Parseq and i_u and i_v the sample indexes.

The ratio distribution has already attracted attention in sample survey and many other areas. Multiple approximation were proposed, either from large sample or hypothesizing a Gaussian distribution of the variables. A more general approach was also proposed [Fieller, 1954] to derive confidence intervals.

Here, we also analyse the results obtained with an approximation of the posterior distribution of the ratio $r = \frac{u}{v}$ build on the hypothesis that the posteriors on expression levels u and v can be well approximated by a gamma distribution. Namely, $\ell \sim \gamma(\kappa_\ell, \theta_\ell), \quad \ell = \{u, v\}$, where the parameters κ_ℓ and θ_ℓ represent the shape and the scale

parameters of the gamma distribution. In practice, the examination of the posterior distributions suggests that this assumption is roughly justified in all of our data-sets for positions that do not have a significant mass at 0. Rescaling v by $\frac{\theta_u}{\theta_v}$ brings the two gamma distributions to the same scale while keeping the shapes unchanged allowing an explicit form of the ratio distribution. The ratio $\tilde{r} = \frac{u}{v} \cdot \frac{\theta_v}{\theta_u}$ has a Beta prime distribution $\mathcal{B}'(\kappa_u, \kappa_v)$ with density

$$\pi(\tilde{r}) \quad = \quad \frac{\tilde{r}^{\kappa_u - 1} \cdot (1+\tilde{r})^{-(\kappa_v + \kappa_u)}}{\beta(\kappa_u, \kappa_v)}, \tag{5.1}$$

where β refers to the beta function. The parameters κ and θ can be estimated for each individual posterior using the method of the moments or by maximum likelihood. By default we use the moment estimates of κ and θ which gives $\hat{\kappa}_\ell = \bar{u}_\ell^2 / \hat{\sigma}_\ell^2$ and $\hat{\theta}_\ell = \hat{\sigma}_\ell^2 / \bar{u}_l$, where $\hat{\sigma}_\ell^2$ and $\hat{u}_\ell$ are the sample estimates for the variance and mean.

We approximate the fold change and the differential expression above a given threshold c by using these three estimation methods:

1. **RM** - the point estimate based on the ratio of posterior means $\hat{r}_{\mathrm{RM}}$

2. **DR-e** - the posterior distribution of the ratio r as approximated by its empirical distribution;

3. **DR-β'** posterior distribution of the ratio r as derived from the Beta prime approximation $\tilde{r} \sim \mathcal{B}'(\kappa_u, \kappa_v)$.

5.2 The Estimation of Differential Expression

At position level we derive the DE at a given fold change c from the cumulative probability above c (tail function). We denote by d_t the probability of DE for the position t and we compute it for each method as follows

$$\textbf{RM: } \mathbf{1}_{\{\hat{r}_{\mathrm{RM}} \geq c\}}; \textbf{ DR-e: } \frac{1}{N^2} \sum_{i_{u,v}}^{1:N^2} \mathbf{1}_{\{\frac{u^{iu}}{v^{iv}} \geq c\}} \text{ and } \textbf{DR-}\beta'\textbf{: } \int_{c\frac{\theta_v}{\theta_u}}^{\infty} \beta'_{\kappa_u,\kappa_v}(r)\,\mathrm{d}r.$$

In a similar manner we can determine positions having a given fold change c. We define a precision level and build a precision interval $[c_1, c_2]$ around the target fold value. We then identify positions with point estimators in this interval (for RM) or with a cumulative probability $P(c_1 \leq r \leq c_2)$ greater than a probability threshold (for DR-e and DR-β').

Read coverage variability induces uncertainty in the estimation of the expression level, which in turn can lead to discontinuities in the annotation of DE regions.

In order to reconstruct **continuous regions with high DE probability**, we use an *ad hoc* method inspired from the local score strategy. We aim to determine continuous regions where locally the average DE probability d_t value is above a given threshold. We use this local approach to account for possible gaps in the DE profile signal resulting local local differences in coverage between the two conditions that are two short to have a biological meaning. We compute a local score ρ_t according to the recursion

$$\rho_t = \min(1, \max(\rho_{t-1} + d_t - m, 0)),$$

where m is the DE probability threshold. We choose first $[t_0 : t'_1]$ segments with $\rho_{t_0-1} = 0$, $\rho_{t_0:t'_1} > 0$ and $\rho_{t'_1+1} = 0$. Then we perform a backward recursion on $[t_0 : t'_1]$ using the same formula $\rho_t = \min(1, \max(\rho_{t+1} + d_t - m, 0))$ and select the region $[t_0 : t_1]$ where $\rho_{t_1+1} = 0$ and $\rho_{t_0:t_1} > 0$. The second step is done to assure reversibility in region selection. In contrast to the classic local score approach, we limit the ρ_t to 1 to avoid effects related to the region length (which is proportional to the cumulative DE probability).

5.3 Evaluation of Differential Expression Estimation

The difficulty raised by evaluating our strategy on real data motivated the use of synthetic datasets. We use synthetic data to evaluate the detection of DE at base-pair resolution and of DE regions. The relevance on real cases is shown on the detection of DE positions.

5.3.1 Synthetic and Real Data-sets

Currently available RNA-Seq simulators (simNGS[Massingham, 2011], Flux simulator) do not account for the coverage variability that we observe on real datasets (section 3.4). Thus, we decide to generate synthetic data by taking transcript expression values from *S. cerevisae* 1 data-set and generating counts according to a dispersion estimated by Parseq on real datasets.

We simulate data for the first 6 chromosomes of *S. cerevisiae* using transcripts from the SGD annotation [Cherry et al., 2012]. For the "wild" data set ($\mathbf{v}$) we set the expression level for each transcript to the value computed from real data (*S. cerevisae* 1). This value is obtained by averaging the counts of reads corresponding to each transcript. For the "mutant" data set ($\mathbf{u}$) we use the same expression levels but we over expressed randomly 15% of the transcripts (corresponding to 200 transcripts) with folds change values of 1/4, 2, 4 or 8. To increase resemblance to real data we integrate local coverage scaling (s, section 3.2). To do this we generate a local scaling profile using the parameters α_s and κ_s estimated by Parseq and scale the count expectation. Conditioning on the expression profile and local scaling we sample read counts according to a Negative Binomial $y_{\ell,t} \sim \mathcal{NB}(s_{\ell,t} \cdot \ell_t, \phi)$, where $\ell \in \{u, v\}$, $s_{\ell,t}$ and ℓ_t are the local scaling and the expression level at position t and ϕ is the over-dispersion ($\phi = 2$ in the simulated data-sets).

We also include in our evaluation data from a study on regulatory non-coding RNAs, Xrn1-sensitive unstable transcripts (XUTs), in *S. cerevisae* [Dijk et al., 2011]. XUTs accumulate in the mutant condition and their loci thus correspond to DE regions. We compare a mutant condition *S. cerevisiae* 4 and a wild condition *S. cerevisiae* 1 (data-set description in table 3.1).

We then run Parseq to estimate the expression profile for both data sets and obtained 2 samples of expression trajectories u^i and v^i, i=1:N. For each condition we run 2200 Parseq sweeps with a thinning step of 10 and we discard the 200 sweeps burn-in.

Results using Parseq estimates were systematically compared with the estimation based on a sliding 100 bp window average of the read counts (SW). In order to avoid border effect (border smoothing), the SW estimate was constrained to the regions covered by at least one read.

For comparison at bp level we consider those positions where Parseq estimated average levels and SW values are above a background value (here 0.01 reads / bp). Reconstruction of DE regions included all values and we set to the background value all expression values below it. Estimation of parameters for the fold change distribution is done as described in the methods. However, in the cases where the degeneracy of the particles lead to an underestimation of the variance we threshold the coefficient of variation c_v to 0.001 and then recalculate the variance: $\hat{\sigma}_{\ell t} = \ell_t \cdot c_v$.

Given a level of fold change c, the results are assessed from three different standpoints: detection of positions with c fold change, of positions with at least c fold change, and the detection of DE regions of level c or above.

Of main relevance, the comparison of results obtained using RM on one side, and DR-e and DR-β' on the other, motivate the choice of approximating the expression level distribution.

5.3.2 Evaluation of Fold Change Estimation

We consider a $\pm$ 25% precision around the correct fold change and threshold the cumulative probability (% of ratio values falling in the precision interval) at 0.3 in order to call DE positions for DR-e and DR-β'.

Increasing this threshold will provide very high PPV but with significant sensitivity loss while, in reverse, at lower thresholds sensitivity can reach 1 but with very low PPV. All results based on Parseq expression level estimations are significantly better in both sensitivity and PPV that those obtained using SW (table 5.1). While DR-e and DR-β' show high sensitivity values for a moderate PPV decrease comparing to RM, the DR-β' seems to achieve a better trade-off between these two indicators.

	DR-e		**DR-β'**		**RM**		**SW**	
Fold	Sens.	PPV	Sens.	PPV	Sens.	PPV	Sens.	PPV
2	0.93	0.08	0.75	0.13	0.51	0.18	0.34	0.09
4	0.86	0.44	0.73	0.55	0.46	0.61	0.32	0.31
8	0.89	0.52	0.70	0.63	0.45	0.70	0.35	0.51

Table 5.1: Detection of change magnitude at position resolution. Synthetic data results. Positions expressed in any dataset lower than 0.01 reads/bp were disregarded. We show sensitivity and positive predictive values. Three fold values (2, 4, 8) were evaluated (precision of ± 25% and cumulative probability threshold of 0.3).

5.3.3 Evaluation of DE Calling at Base-Pair Precision

We discuss first results obtained for the synthetic data-sets and then, we show also results on real data. For the synthetic data, the estimation was done at bp precision for thresholds ranging from 2-fold to 8-fold. RM method performs better than SW mainly in terms of positive predictions (figure 5.1 left panel). DR-e and DR-β' results depend on the probability threshold. High sensitivity values are obtained by lowering the cumulative probability threshold to 0.25 with the cost of having PPV values similar to the SW method. It is important to notice the similar behaviour of the DR-e and DR-β' which sustains the choice of the gamma distributions in modelling the expression level.

We also evaluate (figure 5.1 right panel) the accuracy of DE calling at position resolution in a comparison between two real data-sets *S. cerevisiae* 4 (mutant) and *S. cerevisiae* 1 (wild). As in [Dijk et al., 2011], we scaled the reconstructed mutant expression profile such that levels of tRNA and snoRNA is equal between the two data sets and we excluded already annotated regions [Cherry et al., 2012] from the DE analysis. To minimize the detection of UTRs we also excluded an additional 100 bp on both sides of each annotated gene. The positive predictive values are similar for all DE thresholds and methods. On the contrary, the sensitivity is decreasing with increasing DE thresholds suggesting that XUT detection needs to be done at low (approx 2) fold change thresholds. For given thresholds above $c = 4$ the sensitivity can be increased significantly for the DR-e and DR-β' methods by calling DE positions with a low cumulative probability threshold (0.25)

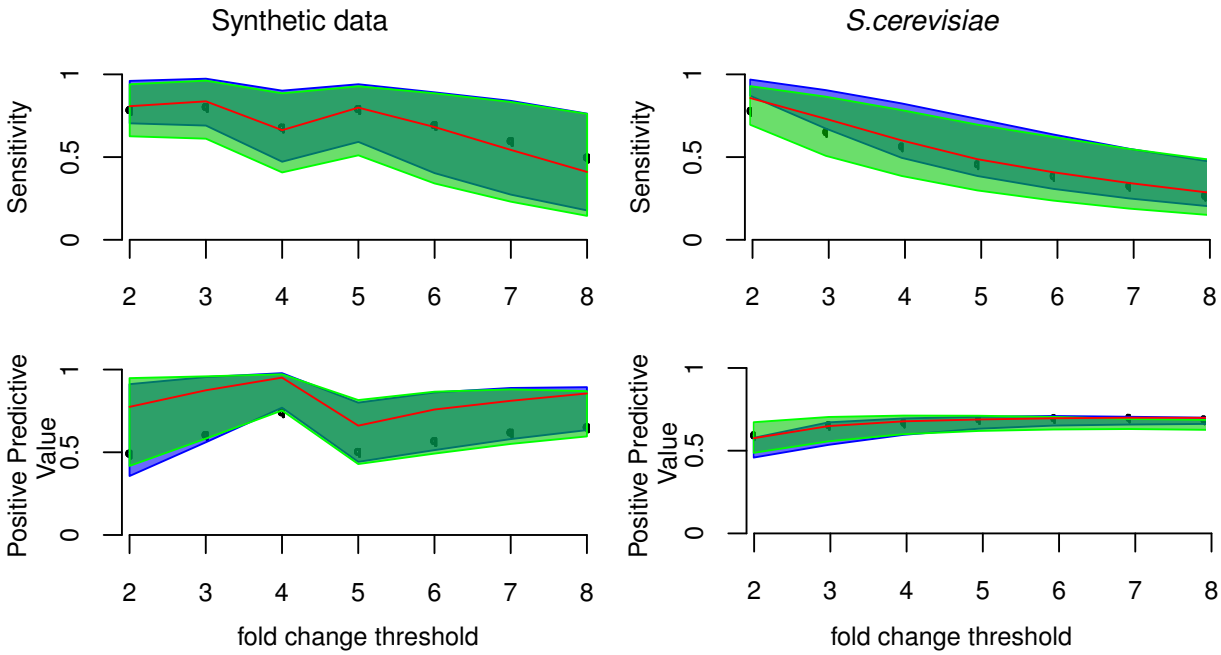

Figure 5.1: Detection of DE at position level for synthetic (left) and real (right) datasets. X-axis: DE threshold. Y-axis: Sensitivity (top) and PPV (bottom). Methods: DR-e (blue band), DR-β' (green band), RM (red line) and SW (black dots). Borders for DR-e and DR-β' bands: Sensitivity - top and low represent the 0.25 and 0.75 cumulative probability thresholds; PPV - top and low represent 0.75 and 0.25 thresholds.

without significant PPV loss.

5.3.4 Calling of DE Regions

Finally, we evaluate DE region detection on synthetic datasets. We compare the borders of DE reconstructed units against those of transcripts with simulated fold change above the same threshold. Sensitivity reaches values above 50% for all methods and most DE thresholds (table 5.2). Parseq based approaches have a net improvement in PPV with DR-e and DR-β' having slightly higher sensitivity than RM. We illustrate the reconstructed DE units in figure 6.2.

	DR-e		**DR-β'**		**RM**		**SW**	
Fold	Sens.	PPV	Sens.	PPV	Sens.	PPV	Sens.	PPV
≥ 2	0.72	0.20	0.72	0.25	0.69	0.25	0.66	0.12
≥ 4	0.57	0.43	0.58	0.39	0.54	0.39	0.63	0.26
≥ 8	0.43	0.39	0.48	0.38	0.38	0.35	0.44	0.23

Table 5.2: Accuracy in 5'-end detection of DE regions for a 50 bp distance cut-off. Results are shown for 3 values of DE thresholds. Estimated DE regions below 100 bp were discarded.

We did not include in the evaluation of calling DE (at base-pair precision or at region level) the results obtained by using DER Finder [Frazee et al., 2014] as this method, which follows the same approach, i.e. reconstructs DE regions from DE at base-pair resolution, was available after the publication of our own. We plan to include its results in further evaluation studies.

Chapter 6

Conclusions and Perspectives

We presented in this thesis a model-based approach for analysing the RNA-Seq read count profiles along the genome. Our work aimed at the reconstruction of the transcription profile from one RNA-Seq data-set and at the estimation of regions with different expression between two conditions. This led to the development of two algorithms: Parseq [Mirauta et al., 2014] and Pardiff [Mirauta et al., 2013].

Parseq implements a statistical approach to estimate the local transcription levels and to identify transcript borders without making use of existing annotation. This transcriptional landscape reconstruction relies on a state-space model to describe transcription level variations in terms of abrupt shifts and more progressive drifts. A compound distribution of read counts was developed to capture the characteristics of RNA-Seq data.

Pardiff describes a method to reconstruct the regions having significant changes in expression between two conditions without making use of predefined annotation and data sets replicates. This method is based on estimates of DE at position level and mitigates the lack of replicates by accounting for the uncertainty in expression level estimation.

From a methodological standpoint our work also demonstrates the feasibility of analysing genome-scale data within the framework of state-space models. We summarize here our contribution to modelling the RNA-Seq data and discuss possible extensions.

6.1 A New RNA-Seq Read Count Model

In section 3.2 we developed a new model for the distribution of RNA-Seq read counts within regions with constant expression level (transcripts). We believe that this will allow on one hand to better estimate the credibility intervals for the expression level and transcription breakpoints, and on the other hand a better quantitative evaluation of the characteristics of RNA-Seq data-sets. In particular, the model addresses amplification effects and local correlation of counts by incorporating corresponding parameters.

For the data-sets (e.g. *S. cerevisiae* 1 and *E. coli* 2) where the amplification parameters were estimated to significant values a comparison to a Parseq model relying on a NB emission revealed a net improvement in the specificity of breakpoints at similar sensitivity and more realistic credibility intervals (in the sense that they are compatible with constant expression along the transcripts).

We expect that a more comprehensive study, including data-sets from various protocols, may reveal associations between protocol characteristics and the values of the parameters from the read count model. Such an association might facilitate the choice of the protocol according to the specific experimental task. During the preliminary analysis of the parameter we encountered a few specific questions. Could the excess of counts in 1 of the isolated counts distribution reveal possible shortcomings of the amplification step? More generally we pose the question of their origin: are they caused by DNA contamination, by antisense artefacts or by background transcription? A second question is related to the local count correlations, a bias with a significant influence on the reconstruction of transcription units. We chose to incorporate it in our model through a variable s that scales the number of initial cDNA molecules but in theory we could build a model that accounts for correlation in subsequent steps (like PCR amplification). In order to detect possible causes of local scaling we started an investigation on the correlation of read counts with the secondary structure. Our first results revealed that such correlation exists but it is mainly explained by a controlling variable (the G and C sequence content). It would be interesting to extend this analysis to study correlations of the local scaling variable with the secondary structure, the GC local content and other sequence related bias sources.

We also show other results concerning the bias due to sequence and position relative to the transcript boundaries. These results confirm already published work and support the

approach with shift and drift, and the incorporation of the local scaling variable. These aim to account for technological artefacts and tackle the problem of modelling the bias. The bias analysis confirmed also that local correlation and other types of bias (figures 3.1 and 3.2) are highly similar for the same protocol conditions (as also stated in [Khrameeva and Gelfand, 2012])

6.2 Designing State Space Models for Genome Wide Analysis

We have built a SSM aiming at the reconstruction of the expression profile. Besides the expression level, the SSM incorporated the variable s to account for local correlations. While the two variables have confounding effects on the read count we can disentangle their trajectories making use of the different tempo of their longitudinal dynamics (short vs. long range). The design of the transition models aimed to distinguish between the two hidden profiles. For x we developed a transition kernel that accounts for 'no changes', smooth changes and shifts in level and for s we designed a piece-wise kernel with a Gamma of mean 1 stationary distribution. In previous versions we estimated the values of the local correlation parameters along with the other parameters within the PG algorithm. In our later implementation we used pre-computed values (section 3.3). The success of distinguishing those two effects is an important illustration of the potential of performing analysis in SSM framework.

It is worth mentioning that the Dirac masses, which account in the transition kernels for maintaining the same level, pose some implementation difficulties. While we can avoid these problems by alternative designs, that account only for smooth changes and shifts, the Dirac masses prove to be computationally efficient and seems relevant from a biological point of view.

A SSM can be designed on a complex Bayesian hierarchy. For example we can incorporate in the SSM variables accounting for missing data (e.g. for positions where reads were discarded due to multiple mapping on the genome). For this we implemented the option of incorporating a variable m_t that takes the value 0 if the position t is in a repeated region and 1 otherwise. The emission writes $e(y_t; x_t, s_t, m_t)$ and takes the value

1 if $m_t = 0$ and $e(y_t; x_t, s_t)$ otherwise. In principle, we could also design SSMs for the genome wide analysis from multiple datasets eventually generated with different protocols (e.g. global and 5'-end sequencing). Such models might aim to capture alternative transcription breakpoints or to return breakpoint consensus, to directly approximate DE positions or to return a more robust reconstruction of the local correlation profile.

For data produced with the same RNA-Seq protocol (that should have a high similarity in bias) we propose a SSM (DAG in figure 6.1) aiming at the identification of alternative transcription breakpoints. Slight changes in the design (introducing x_{de} variable to control for DE at position t and considering the same breakpoints for the two data-sets) can lead to a SSM aiming at detecting the differences in expression.

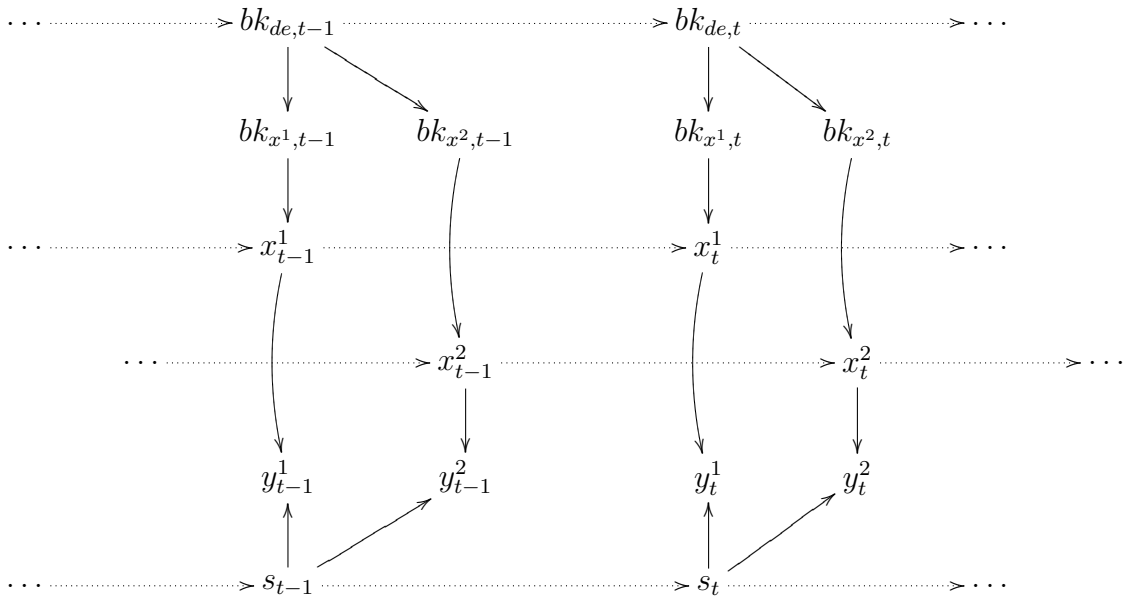

Figure 6.1: DAG for SSM for two data-sets. The variables y^1,x^1, bk_{x^1} and respectively y^2,x^2, bk_{x^2} represent the counts, expression levels and breakpoints for the two data-sets. The local correlation s is the same in the two data-sets. The variable bk_{de} accounts for the alternative usage of the breakpoints, $bk_{de,t} = \mathbf{1}_{\{bk_{x^1,t}=bk_{x^1,t}\}}$. For simplicity we do not show parameters as in the augmented state space described in figure 4.1.

6.3 Applying Sequential Monte Carlo Methods on Genome-Wide Scale

Our choice for the Particle Gibbs algorithm with backward sampling was instrumental in obtaining the transcription profile reconstruction. The PG algorithm, like most Monte Carlo approaches, is computationally intensive. We reduced the overall computation cost by combining the optimisation of proposals and backward sampling of trajectories (that allowed approximations using a lower number of particles and PG sweeps). To confirm the benefits of our choice of the PG algorithm we performed an analysis of the accuracy of this algorithm and of the Sequential Importance Resampling. We present empirical results in section 4.2.4 for several algorithm set-ups and various proposals. These illustrate that accurate estimation is reached by PG (using CSMC) in most scenarios and even for a small number of particles (P). The trajectory update using the SIR algorithm instead of CSMC shows biased estimations for unfitted proposals and practical numbers of particles. We observed also that CSMC and SIR require backward sampling for long sequences to tackle the particle coalescence issue. By a simple rule of thumb the size of the update trajectory should not be much longer than P if only filtering is used.

Aiming at improving particle mixing we searched for efficient proposals. We developed a mixture proposals with fixed weights but position dependent components. Next, based on an adaptive strategy, the Population Monte Carlo, we tuned the proposal by optimizing the weights to get closer to $\pi(x_t \mid x_{t-1}, y_{1:T})$, the kernel of the heterogeneous Markov chain $\pi(x_{1:T} \mid y_{1:T})$ we want to reconstruct. With this adaptive strategy we manage to improve particle mixing within Particle Gibbs sweeps and this improvement proves to be more important for low number of particles (section 4.2.5). In the algorithms presented in this thesis the proposals have genome-wide constant weights. In the future we may want to further optimize the proposal and estimate mixture weights that could differ between positions. We believe that these may allow a further decrease in the number of particles.

Proposal tuning may answer a more practical question we asked ourselves: is the proposal optimisation for filtering updates (aiming at $\pi(x_t \mid x_{t-1}, y_t)$) the best way to achieve global smoothed approximations or is it outperformed by proposals that generate a sample close to the kernel of the "smoothed" values $\pi(x_t \mid x_{t-1}, y_{1:T})$?

The adaptive proposal can be complemented by an adaptive setting of the number

of particles used for a position. Indeed, on positions where proposals sample values approximate well the target it is not needed to use a high sample size. The definition of a criteria for setting the number of particles is related to the proposal fit and may rely on its evaluation by the ESS or KL divergence (see also [Fox, 2001]).

A last point we want to underline in this section is related to the number of trajectories reconstructed at each PG. In the algorithms we present (for e.g. algorithm 11 we backward sample one trajectory. Backward sampling can reconstruct several trajectories and respectively obtain several correlated samples from the marginal distributions $\pi(x_t \mid y_{1:T})$ that can be used for improve the efficiency of the algorithm in estimating the credibility intervals.

6.4 Results in Estimating the Transcription and DE Profiles

The algorithms we developed, Parseq and Pardiff, permit in-depth analysis at genome-wide scale of RNA-Seq data. The running time does not depend much on the depth of the sequencing, but is proportional to genome length, which makes them more suited to microbial genomes. For long and less condensed genomes, the computation time would greatly benefit from using an adaptive number of particles. However, while the memory print is kept to an almost constant value, the current implementation needs significant changes in the structure of the output data to be compatible with long (10^9) genomes.

From RNA-Seq datasets, Parseq algorithm in conjunction with Pardiff can provide a wide range of results including both point estimators and credibility intervals. Notably, direct results include for one data-set the approximations of genome-wide expression level, the estimations of transcript breakpoints and of the transcript probability at base-pair resolution. For two conditions, these results are complemented by the approximation of fold change and estimation of DE probability at base-pair resolution. The reconstructed profiles give also insights on the transcriptional units, transcription regions and DE regions. We illustrate the possible results that can be obtained by using Parseq and Pardiff in figures 4.5 (for one data-set) and 6.2 (for two data-sets).

Some biological questions and notably those involving correlation studies of the ex-

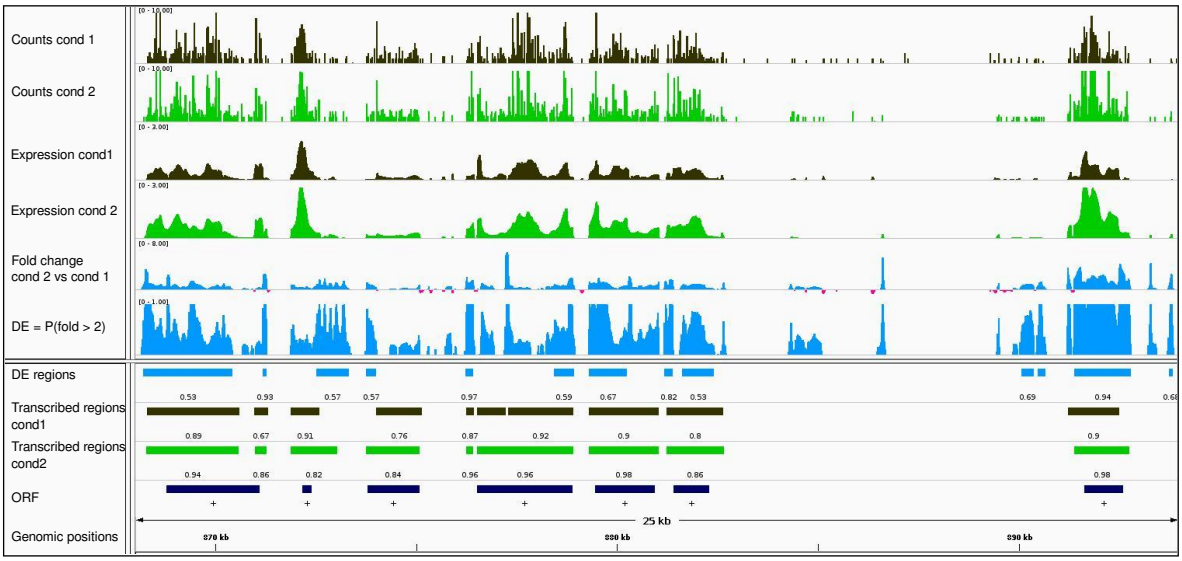

Figure 6.2: The transcribed and DE results obtained with Parseq and Pardiff on a region of 25kbp (chromosome 5, plus strand) from two data-sets of *C. albicans* - 1 (brown) and 2 (green). Top lanes: 5' counts; expression profile; average fold change and DE probability for a DE threshold of 2 (blue lanes). Bottom lanes: DE (blue) and transcription (brown and green) units reconstructed with the *ad hoc* method (section 5.2) from the DE and transcription probabilities (average unit probability displayed bellow); ORF (dark blue).

pression with various genome-wide data (replication origins, methylation profiles, histone maps, etc.) can be best answered considering the genome wide expression profiles. Also, the genome wide profiles could be integrated in a more natural fashion with results of various protocols or obtained on different data-sets. For both cases, the availability of confidence scores and credibility intervals could be relevant for more robust analysis.

Other biological questions require the determination of expression level or DE at transcription unit level. For such studies, the RNA-Seq data may be used to build a reference annotation as a part of a compendium of experiments (as done from tiling array data [Nicolas et al., 2012]) or to complete, as in [Lin et al., 2013], results from protocols targeting more specifically the sequencing of transcript ends (CAGE, dRNA-Seq, TIF-Seq [Pelechano et al., 2013], etc.). While it greatly facilitates the genetic analysis, the association of one genomic region to one transcription unit does not capture the real diversity of transcriptomes. Direct isoform quantification is possible by mate pair sequencing of transcript ends or by sequencing the complete sequences of initial mRNAs but these technologies are not yet popular. The expression level changes within expressed regions might

provide valuable insights for the identification of boundaries for dominant isoforms.

Historically, the sequencing protocols kept the pace with technological advancements in microscopy, biochemistry and computation. Tuning conditions and materials will most likely generalize protocols aiming at direct RNA sequencing methods (like [Ozsolak et al., 2009]) and single-molecule sequencing (like [Eid et al., 2009]). While it is unlikely that randomness and noise will be eliminated we believe that the current dominant bias (pertaining to sequence content) will be mitigated. We believe that, whatever the technological evolutions may be, the SMSs can be the answer to a wide range of problems pertaining to genome-wide analysis. The recent developed PMCMC algorithms are valuable tools for obtaining precise approximations even in complex SSMs.

Bibliography

D. Aird, M. G. Ross, W.-S. Chen, et al. Analyzing and minimizing PCR amplification bias in illumina sequencing libraries. *Genome Biology*, 12(2):R18, 2011.

S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, 2010.

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

A. Barski, S. Cuddapah, K. Cui, et al. High-resolution profiling of histone methylations in the human genome. *Cell*, 129(4):823 – 837, 2007.

Y. Benjamini and T. P. Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, 40(10):e72–e72, 2012.

S. Berget, C. Moore, and P. Sharp. Spliced segments at the 5' terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences*, 74, 1977.

P. Bertone, V. Stolc, T. E. Royce, et al. Global identification of human transcribed sequences with genome tiling arrays. *Science*, 306(5705):2242–2246, 2004.

R. Bohnert and G. Ratsch. rQuant.web: a tool for RNA-seq-based transcript quantitation. *Nucleic Acids Research*, 38(Web Server):W348–W351, 2010.

B. Bolstad, R. Irizarry, M. strand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

A. Borries, J. Vogel, and C. M. Sharma. Differential RNA sequencing (dRNA-seq): Deep-sequencing-based analysis of primary transcriptomes. In *Tag-Based Next Generation Sequencing*. Wiley-VCH, 2012.

C. Breto, D. He, E. L. Ionides, and A. A. King. Time series analysis via mechanistic models. *The Annals of Applied Statistics*, 3(1):319–348, 2009.

J. Bullard, E. Purdom, K. Hansen, and S. Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC Bioinformatics*, 11(1):94, 2010.

O. Cappe, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer, 2005.

O. Cappe, R. Douc, and A. Guilin. Adaptive importance sampling in general mixture classes. *arXiv:0710.4242*, 2007.

Y. d. Carafa, E. Brody, and C. Thermes. Prediction of rho-independent escherichia coli transcription terminators. a statistical analysis of their RNA stem-loop structures. *Journal of Molecular Biology*, 216(4):835–858, 1990.

G. Celeux, D. Chauveau, and J. Diebolt. On stochastic versions of the EM algorithm. Technical Report 2514, INRIA, 1995.

Z. Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

J. M. Cherry, E. L. Hong, C. Amundsen, et al. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic Acids Res*, 40(Database issue):D700–D705, 2012.

N. Chopin and S. S. Singh. On the particle gibbs sampler. *arXiv:1304.1887*, 2013.

A. Cleynen, S. Dudoit, and S. Robin. Comparing segmentation methods for genome annotation based on RNA-seq data. *Journal of Agricultural, Biological, and Environmental Statistics*, 19(1):101–118, 2014.

A. Cleynen, T. M. Luong, G. Rigaill, and G. Nuel. Fast estimation of the integrated completed likelihood criterion for change-point detection problems with applications to next-generation sequencing data. *Signal Processing*, 98:233–242, 2014.

F. Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.

J. Dekker, K. Rippe, M. Dekker, and N. Kleckner. Capturing chromosome conformation. *Science*, 295(5558):1306–1311, 2002.

E. L. v. Dijk, C. L. Chen, Y. d'Aubenton Carafa, et al. XUTs are a class of xrn1-sensitive antisense regulatory non-coding RNA in yeast. *Nature*, 475(7354):114–117, 2011.

M.-A. Dillies, A. Rau, J. Aubert, et al. A comprehensive evaluation of normalization methods for illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics*, 14(6):671–683, 2013.

R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert. Convergence of adaptive mixtures of importance sampling schemes. *The Annals of Statistics*, 35(1):420–448, 2007.

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

J. Eid, A. Fehr, J. Gray, et al. Real-time DNA sequencing from single polymerase molecules. *Science*, 323(5910):133–138, 2009.

E. Evguenieva-Hackenberg and G. Klug. New aspects of RNA processing in prokaryotes. *Current Opinion in Microbiology*, 14(5):587–592, 2011.

E. C. Fieller. Some problems in interval estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 16(2):175–185, 1954.

D. Fox. KLD-sampling: Adaptive particle filters. In *Advances in neural information processing systems*, 2001.

C. Francke, T. G. Kormelink, Y. Hagemeijer, et al. Comparative analyses imply that the enigmatic sigma factor 54 is a central controller of the bacterial exterior. *BMC genomics*, 12(1):385, 2011.

A. Frank. The biochemistry of the nucleic acids, purines, and pyrimidines - annual review of biochemistry, 10(1):221. *Annual Review of Biochemistry*, 10:221–244, 1941.

A. C. Frazee, S. Sabunciyan, K. D. Hansen, R. A. Irizarry, and J. T. Leek. Differential expression analysis of RNA-seq data at single-base resolution. *Biostatistics*, 15(3):413–426, 2014.

J. Gilbert and M. Hughes. *Gene Expression Profiling: Metatranscriptomics*, volume 733 of *Methods in Molecular Biology*. Humana Press, 2011.

S. J. Godsill, A. Doucet, and M. West. Monte carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, 2004.

S. R. Goldman, R. H. Ebright, and B. E. Nickels. Direct detection of abortive RNA transcripts in vivo. *Science*, 324(5929):927–928, 2009.

N. Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Radar and Signal Processing, IEEE*, volume 140, 1993.

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

M. Guttman, M. Garber, J. Z. Levin, et al. Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nature Biotechnology*, 28(5):503–510, 2010.

N. R. Guydosh and R. Green. Dom34 rescues ribosomes in 3 untranslated regions. *Cell*, 156(5):950 – 962, 2014.

K. D. Hansen, S. E. Brenner, and S. Dudoit. Biases in illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Research*, 38(12):e131–e131, 2010.

J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, IEEE*, 2006.

Y.-F. Huang, S.-C. Chen, Y.-S. Chiang, T.-H. Chen, and K.-P. Chiu. Palindromic sequence impedes sequencing-by-ligation mechanism. *BMC Systems Biology*, 6(Suppl 2): S10, 2012.

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

E. L. Ionides, C. Breto, and A. A. King. Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 103(49):18438–18443, 2006.

T. Jackson, R. Spriggs, N. Burgoyne, C. Jones, and A. Willis. Evaluating bias-reducing protocols for RNA sequencing library preparation. *BMC Genomics*, 15(1):569, 2014.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

M. Kanamori-Katayama, M. Itoh, H. Kawaji, et al. Unamplified cap analysis of gene expression on a single-molecule sequencer. *Genome Research*, 21(7):1150–1159, 2011.

D. Karlis and E. Xekalaki. Mixed poisson distributions. *International Statistical Review*, 73(1):35–58, 2005.

M. J. Kazmierczak, M. Wiedmann, and K. J. Boor. Alternative sigma factors and their roles in bacterial virulence. *Microbiology and Molecular Biology Reviews*, 69(4):527–543, 2005.

M. Kertesz, Y. Wan, E. Mazor, et al. Genome-wide measurement of RNA secondary structure in yeast. *Nature*, 467(7311):103–107, 2010.

E. E. Khrameeva and M. S. Gelfand. Biases in read coverage demonstrated by interlaboratory and interplatform comparison of 117 mRNA and genome sequencing experiments. *BMC Bioinformatics*, 13(Suppl 6):S4, 2012.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500): 1590–1598, 2012.

F. Krueger, B. Kreck, A. Franke, and S. R. Andrews. DNA methylome analysis using short bisulfite sequencing data. *Nature Methods*, 9(2):145–151, 2012.

B. Langmead, C. Trapnell, M. Pop, and S. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

I. Lasa, A. Toledo-Arana, A. Dobin, et al. Genome-wide antisense transcription drives mRNA processing in bacteria. *Proceedings of the National Academy of Sciences*, 108 (50):20172–20177, 2011.

M. Lavielle and G. Teyssire. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, 46(3):287–306, 2006.

Y. S. Lee, K. Nakahara, J. W. Pham, et al. Distinct roles for drosophila dicer-1 and dicer-2 in the siRNA/miRNA silencing pathways. *Cell*, 117(1):69–81, 2004.

M. M. Leimena, J. Ramiro-Garcia, M. Davids, et al. A comprehensive metatranscriptome analysis pipeline and its validation using human small intestine microbiota datasets. *BMC Genomics*, 14(1):530, 2013.

J. Li, H. Jiang, and W. Wong. Modeling non-uniformity in short-read rates in RNA-seq data. *Genome Biology*, 11(5):R25, 2010.

S. Li, X. Dong, and Z. Su. Directional RNA-seq reveals highly complex condition-dependent transcriptomes in e. coli k12 through accurate full-length transcripts assembling. *BMC genomics*, 14(1):520, 2013.

Y.-F. Lin, D. R. A, S. Guan, L. Mamanova, and K. J. McDowall. A combination of improved differential and global RNA-seq reveals pervasive transcription initiation and events in all stages of the life-cycle of functional RNAs in propionibacterium acnes, a major contributor to wide-spread human disease. *BMC Genomics*, 14(1):620, 2013.

F. Lindsten. An efficient stochastic approximation EM algorithm using conditional particle filters. In *38th International Conference on Acoustics, Speech, and Signal Processing*, 2013.

J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2008.

H. Lodish, A. Berk, S. L. Zipursky, et al. *Molecular Cell Biology*. W.H.Freeman, 4 edition, 2000.

R. Lorenz, S. H. Bernhart, C. H. Zu Siederdissen, et al. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.

J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18(9):1509–1517, 2008.

J. A. Martin and Z. Wang. Next-generation transcriptome assembly. *Nature Reviews Genetics*, 12(10):671–682, 2011.

T. Massingham. simNGS - software for simulating next-gen sequencing data., 2011. Accessed on 2013-11-12.

D. J. McCarthy and G. K. Smyth. Testing significance relative to a fold-change threshold is a TREAT. *Bioinformatics*, 25(6):765–771, 2009.

R. McClure, D. Balasubramanian, Y. Sun, et al. Computational analysis of bacterial RNA-seq data. *Nucleic Acids Research*, 41(14):e140–e140, 2013.

B. Mirauta, P. Nicolas, and H. Richard. Pardiff: Inference of differential expression at base-pair level from RNA-seq experiments. In *New Trends in Image Analysis and Processing ICIAP 2013*, volume 8158 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013.

B. Mirauta, P. Nicolas, and H. Richard. Parseq: reconstruction of microbial transcription landscape from RNA-seq read counts using state-space models. *Bioinformatics*, 30(10): 1409–1416, 2014.

M. Mohle. The time back to the most recent common ancestor in exchangeable population models. *Advances in Applied Probability*, (1):78–97, 2004.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods*, 5(7):621–628, 2008.

P. Nicolas, A. Leduc, S. Robin, et al. Transcriptional landscape estimation from tiling array data using a model of signal shift and drift. *Bioinformatics*, 25(18):2341–2347, 2009.

P. Nicolas, U. Mder, E. Dervyn, et al. Condition-dependent transcriptome reveals high-level regulatory architecture in bacillus subtilis. *Science*, 335(6072):1103–1106, 2012.

S. F. Nielsen. The stochastic EM algorithm: Estimation and asymptotic results. *Bernoulli*, 6(3):457, 2000.

G. Nuel. Effective p-value computations using finite markov chain imbedding (FMCI): application to local score and to pattern statistics. *Algorithms for Molecular Biology*, 1(1):5, 2006.

F. Ozsolak, A. R. Platt, D. R. Jones, et al. Direct RNA sequencing. *Nature*, 461(7265): 814–818, 2009.

F. Ozsolak, P. Kapranov, S. Foissac, et al. Comprehensive polyadenylation site maps in yeast and human reveal pervasive alternative polyadenylation. *Cell*, 143(6):1018–1029, 2010.

E. Park, B. Williams, B. J. Wold, and A. Mortazavi. RNA editing in the human ENCODE RNA-seq data. *Genome research*, 22(9):1626–1633, 2012.

K. D. Passalacqua, A. Varadarajan, B. D. Ondov, et al. Structure and complexity of a bacterial transcriptome. *Journal of Bacteriology*, 191(10):3203–3211, 2009.

V. Pelechano, W. Wei, and L. M. Steinmetz. Extensive transcriptional heterogeneity revealed by isoform profiling. *Nature*, 497(7447):127–131, 2013.

F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin. A statistical approach for array CGH data analysis. *BMC bioinformatics*, 6(1):27, 2005.

F. Picard, E. Lebarbier, E. Budinsk, and S. Robin. Joint segmentation of multivariate gaussian processes using mixed linear models. *Computational Statistics & Data Analysis*, 55(2):1160–1170, 2011.

J. E. Prez-Ortn, P. M. Alepuz, and J. Moreno. Genomics and gene transcription kinetics in yeast. *Trends in Genetics*, 23(5):250–257, 2007.

L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings, IEEE*, volume 77, 1989.

G. Ramaswami, R. Zhang, R. Piskol, et al. Identifying RNA editing sites using RNA sequencing data alone. *Nature Methods*, 10(2):128–132, 2013.

H. Richard, M. H. Schulz, M. Sultan, et al. Prediction of alternative isoforms from exon expression levels in RNA-seq experiments. *Nucleic Acids Research*, 2010.

P. Richard and J. L. Manley. Transcription termination by nuclear RNA polymerases. *Genes & Development*, 23(11):1247–1269, 2009.

J. P. Richardson. Rho-dependent termination and ATPases in transcript termination. *Biochimica et Biophysica Acta (BBA)-Gene Structure and Expression*, 1577(2):251–260, 2002.

G. Rigaill. Pruned dynamic programming for optimal multiple change-point detection. *arXiv:1004.0887*, 2010.

J. L. Rinn and H. Y. Chang. Genome regulation by long noncoding RNAs. *Annual Review of Biochemistry*, 81(1):145–166, 2012.

M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1): 139–140, 2010.

S. Rouskin, M. Zubradt, S. Washietl, M. Kellis, and J. S. Weissman. Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo. *Nature*, 505(7485):701–705, 2014.

H. Salgado, M. Peralta-Gil, S. Gama-Castro, et al. RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic Acids Research*, 41(D1):D203–D213, 2013.

F. Sanger, S. Nicklen, and A. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, (74):5463–7, 1977.

H. Schaller, C. Gray, and K. Herrmann. Nucleotide sequence of an RNA polymerase binding site from the DNA of bacteriophage fd. *Proceedings of the National Academy of Science*, (72):737–741., 1975.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, 1995.

C. M. Sharma, S. Hoffmann, F. Darfeuille, et al. The primary transcriptome of the major human pathogen helicobacter pylori. *Nature*, 464(7286):250–255, 2010.

T. Shiraki, S. Kondo, S. Katayama, et al. Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proceedings of the National Academy of Sciences*, 100(26):15776–15781, 2003.

H. Takahashi, T. Lassmann, M. Murata, and P. Carninci. 5[prime] end-centered expression profiling using cap-analysis gene expression and next-generation sequencing. *Nature Protocols*, 7(3):542–561, 2012.

H. Thorvaldsdottir, J. T. Robinson, and J. P. Mesirov. Integrative genomics viewer (IGV): high-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, 14(2):178–192, 2013.

C. Trapnell, B. A. Williams, G. Pertea, et al. Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, 2010.

C. Trapnell, D. G. Hendrickson, M. Sauvageau, et al. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature Biotechnology*, 31(1):46–53, 2012.

T. van Opijnen, K. L. Bodi, and A. Camilli. Tn-seq: high-throughput parallel sequencing for fitness and genetic interaction studies in microorganisms. *Nature Methods*, 6(10): 767–772, 2009.

P. Vandeputte, S. Pradervand, F. Ischer, et al. Identification and functional characterization of rca1, a transcription factor involved in both antifungal susceptibility and host response in candida albicans. *Eukaryotic Cell*, 11(7):916–931, 2012.

V. Velculescu, L. Zhang, and K. Kinzler. Serial analysis of gene expression. *Science*, 270, 1995.

L. Wan, X. Yan, T. Chen, and F. Sun. Modeling RNA degradation for RNA-seq with applications. *Biostatistics*, 13(4):734–747, 2012.

E. T. Wang, R. Sandberg, S. Luo, et al. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221):470–476, 2008.

K. Wang, D. Singh, Z. Zeng, et al. MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*, 38(18):e178, 2010.

Y. Wang and P. L. deHaseth. Sigma 32-dependent promoter activity in vivo: Sequence determinants of the groE promoter. *Journal of Bacteriology*, 185(19):5800–5806, 2003.

Z. Wang, M. Gerstein, and M. Snyder. RNA-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

S. William Roy and W. Gilbert. The evolution of spliceosomal introns: patterns, puzzles and progress. *Nature Review of Genetics*, 7(3):211–221, 2006.

R. Wu. Nucleotide sequence analysis of DNA. *Nature*, 236:198–200, 1972.

C. Yang, E. Bolotin, T. Jiang, F. M. Sladek, and E. Martinez. Prevalence of the initiator over the TATA box in human and yeast genes and identification of DNA motifs enriched in human TATA-less core promoters. *Gene*, 389(1):52–65, 2007.

M. Yassour, T. Kaplan, H. B. Fraser, et al. Ab initio construction of a eukaryotic transcriptome by massively parallel mRNA sequencing. *Proceedings of the National Academy of Sciences*, 106(9):3264–3269, 2009.

Z. Zhang. Mapping of transcription start sites in saccharomyces cerevisiae using 5' SAGE. *Nucleic Acids Research*, 33(9):2838–2851, 2005.

A. Zymnis, S. Boyd, and D. Gorinevsky. Mixed state estimation for a linear gaussian markov model. In *Decision and Control*, volume 47. IEEE, 2008.