BAYESIAN METHODS FOR GENE EXPRESSION ANALYSIS FROM HIGH-THROUGHPUT SEQUENCING DATA

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2014

By
Peter Glaus
School of Computer Science

Contents

List of Tables

List of Figures

Word Count: 42,180

Abstract

Bayesian Methods for Gene Expression Analysis from High-Throughput Sequencing data

Peter Glaus

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2014

We study the tasks of transcript expression quantification and differential expression analysis based on data from high-throughput sequencing of the transcriptome (RNA-seq).

In an RNA-seq experiment subsequences of nucleotides are sampled from a transcriptome specimen, producing millions of short reads. The reads can be mapped to a reference to determine the set of transcripts from which they were sequenced. We can measure the expression of transcripts in the specimen by determining the amount of reads that were sequenced from individual transcripts.

In this thesis we propose a new probabilistic method for inferring the expression of transcripts from RNA-seq data. We use a generative model of the data that can account for read errors, fragment length distribution and non-uniform distribution of reads along transcripts. We apply the Bayesian inference approach, using the Gibbs sampling algorithm to sample from the posterior distribution of transcript expression. Producing the full distribution enables assessment of the uncertainty of the estimated expression levels.

We also investigate the use of alternative inference techniques for the transcript expression quantification. We apply a collapsed Variational Bayes algorithm which can provide accurate estimates of mean expression faster than the Gibbs sampling algorithm.

Building on the results from transcript expression quantification, we present a new method for the differential expression analysis. Our approach utilizes the full posterior distribution of expression from multiple replicates in order to detect significant changes in abundance between different conditions. The method can

be applied to differential expression analysis of both genes and transcripts.

We use the newly proposed methods to analyse real RNA-seq data and provide evaluation of their accuracy using synthetic datasets. We demonstrate the advantages of our approach in comparisons with existing alternative approaches for expression quantification and differential expression analysis.

The methods are implemented in the BitSeq package, which is freely distributed under an open-source license. Our methods can be accessed and used by other researchers for RNA-seq data analysis.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

First I would like to thank my supervisor Magnus Rattray for great support and mentorship. Thanks for all the ideas and comments, optimism, patience, witty remarks and everlasting good mood.

I also wish to thank Jon Shapiro for backing my research and keeping an eye on me within the Computer Science.

I am very grateful to Antti Honkela, for thoughts, discussions and playing a big role throughout my project. I also appreciate the great collaborations with James Hensman and Panos Papastamoulis.

Many thanks to all my lab mates and friends from FLS, MLO and SITraN that made the stay in Manchester fun and great experience. Thanks to Tomasz, Maria, Panos, Jing, Adam, Richard, Mauricio, Nicolò, Joe, Freddie, Michalis and Jon.

Thanks to my family for all their support and encouragement throughout my 22 years of studying.

Finally, this thesis would never have been finished without the motivation, help, patience, care and support from my girlfriend Baška. I want to thank her for all that and for being part of my life.

Ďakujem.

Chapter 1

Introduction

It took almost 15 years, 20 research centres and billions of dollars to sequence 99% of DNA using Sanger sequencing in the Human Genome Project (Human Genome Sequencing Consortium International, 2004). Since the beginning of the project, advancement of technology produced new, faster and cheaper ways of sequencing. Next Generation Sequencing (NGS) technologies, also referred to as High-Throughput Sequencing (HTS) or Massive Parallel Sequencing, now allow sequencing a human genome in a few weeks, with costs well under one hundred thousand dollars. Apart from the direct application of exploring the genomic code, the high speed and low cost of the new technologies enabled a range of new scientific approaches in genomics, transcriptomics, metagenomics and other related fields.

High-throughput sequencing of the transcriptome (RNA-seq) is one of the major applications of NGS. In an RNA-seq experiment, RNA molecules are extracted from a sample, reverse-transcribed into complementary DNA, which is sequenced by a high-throughput sequencing device. RNA-seq can be used for discovering which transcripts are present in a sample and also for the key task of measuring their expression — the abundance of transcript molecules within a sample. The high reproducibility, large dynamic range and ability to detect novel splice variants make RNA-seq an attractive alternative to previously used technologies such as microarrays. Here we study the problem of transcript expression quantification through sequencing and related challenges of analysis of RNA-seq data.

1.1 Objectives of this thesis

The aim of this thesis is to investigate probabilistic methods for the analysis of RNA-seq data using Bayesian approaches. We focus on the quantification of transcript and gene expression levels using high-throughput sequencing technologies and comparison of abundance estimates between different conditions.

The quantification of transcript expression levels from RNA-seq data cannot be solved exactly in most cases. The process of high-throughput sequencing is a random process that samples small pieces of evidence of the molecules being analysed. Despite the high reproducibility of experiments, the RNA-seq data contains random effects and errors. The similarity of alternative transcript sequences can lead to situations which make the exact quantification difficult or impossible and necessitate probabilistic approaches.

The differential expression analysis is a direct extension of the expression quantification task. To detect effects that are truly caused by various conditions, uncertainty of the abundance estimates as well as natural abundance fluctuations have to be considered within the analysis.

In these kinds of problems, probabilistic methods provide a natural framework for dealing with the uncertainty. We apply the Bayesian approaches which represent variables in the form of probability distributions and provide ways for manipulating the distributions in further analysis.

We summarise the goals of this research project in the following four points:

1. Create a method for quantification of transcript expression from RNA-seq data, which will provide accurate expression estimates as well as a measure of uncertainty for the estimates.

2. Investigate efficient inference algorithms that can be used for estimating transcript expression and accommodate constantly increasing size of RNA-seq data.

3. Study the use of probabilistic methods for detecting abundance changes, while accounting for biological fluctuations and leveraging the uncertainty measure of expression estimates.

4. Provide implementation of the expression quantification and differential expression analysis methods that can be used by bioinformaticians and other researchers performing RNA-seq data analysis.

1.2 Structure of the thesis

In the rest of this chapter we provide an overview of high-throughput sequencing and RNA-seq as well as a review of related approaches used for RNA-seq data analysis. We look at the NGS technologies currently being used, the properties of the data being generated and the applications of high-throughput sequencing. Then we introduce the RNA-seq procedure for sequencing the transcriptome and define the problems of expression quantification and differential expression analysis. We also include a short review of related methods for expression estimation, differential expression analysis and other applications of RNA-seq. The final section provides an overview of the Bayesian principles used in this thesis.

In Chapter 2 we propose a probabilistic approach for transcript expression quantification. We describe the generative model, alignment likelihood calculation and non-uniform read distribution bias correction method used in our approach. Standard and collapsed versions of the Gibbs sampling algorithm are used for inference of the posterior distribution of expression.

In the results section we examine output produced by our approach. We evaluate the accuracy of expression quantification using synthetic data and real data with validation. We also provide comparison of our method with other state-of-the-art methods for expression quantification.

In Chapter 3 we present a novel method for differential expression analysis that builds upon our expression quantification procedure presented in Chapter 2. The differential expression analysis method uses expression estimates from replicated samples to assess the biological variance. The novelty of the method is in the use of entire posterior distribution produced by our quantification method. This enables propagation of uncertainty from the quantification stage into the differential expression analysis results.

We present the workings of our method and also evaluate its performance. We use synthetic data for evaluation as it enables comparison against known ground truth. Comparison with alternative differential expression analysis approaches is provided as well.

In Chapter 4 we investigate the use of alternative inference approaches for the expression estimation problem. Instead of using the Gibbs sampling algorithm, we apply a Variational Bayes approximate inference procedure to estimate the expression levels of transcripts. The alternative inference method provides a high

level of accuracy in terms of mean estimate, with much shorter run time.

We compare the results obtained by the Variational Bayes inference with those produced by Gibbs sampling in order to assess the effectiveness of the approximative inference. We also evaluate the inference method independently using synthetic data with known ground truth expression.

In Chapter 5 we conclude the thesis by summarizing the presented contributions. Furthermore, we outline possible extensions of our work and new approaches building on the methods presented in this thesis.

1.3 High-throughput sequencing

With the aim of reducing cost and increasing throughput of Sanger sequencing, various platforms have been developed that are referred to as next, or second, generation sequencing devices. The improvement was brought by means of great parallelisation of the sequencing process of a single sample.

While every NGS platform uses specific mechanisms during the sequencing process, they all share common traits of high-throughput sequencing. The most important one is that instead of producing one complete sequence of analysed sample, the output consists of millions of short reads, or tags. The reads are randomly sampled nucleotide-subsequences of the original molecules present in the sample. The length of the reads is shorter than those produced by Sanger sequencing, ranging from 25 base pairs (bp) up to $400bp$, with only the newest pyrosequencing devices being able to produce reads of length $700bp$. This property of the high-throughput sequencing technologies output poses new challenges for analysis of this kind of data.

The ability to sequence vast amounts of molecules at low cost prompted researchers to experiment with applying NGS to a wide variety of tasks. These range through *de-novo* genome assembly, nucleotide variation detection, protein binding analysis and transcriptome quantification.

Additionally, NGS revolutionised the field of Bioinformatics, the novel applications of sequencing as well as the short-read high-throughput properties of the data require new computational approaches for the analysis. First of all, the shear amount of generated data would make any kind of analysis impossible without modern computing technologies. Processing this kind of datasets can be time consuming as well as storage intensive. Secondly, the sequence being

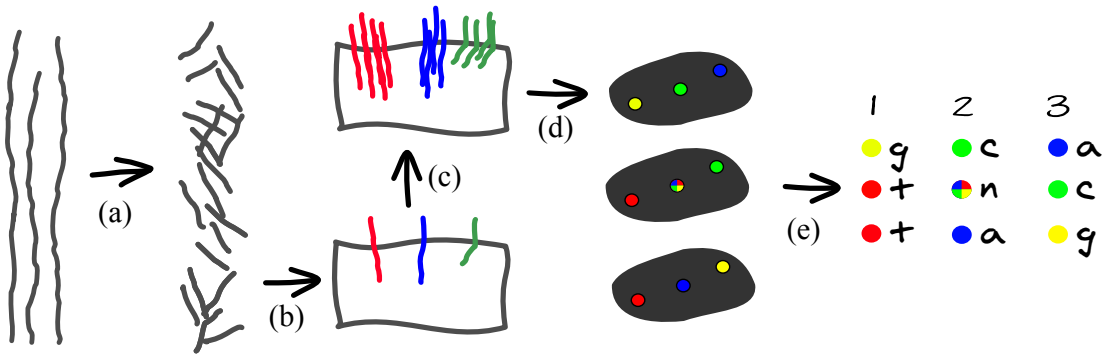


Figure 1.1: **Outline of high-throughput sequencing protocol** In the first phase of the protocol: molecules are fragmented and size selected (a), the fragments are attached to a carrier such as a glass slide (b) and the attached molecules are amplified through polymerase chain reaction (PCR) (c). In the second phase consisting of multiple cycles, fluorescent markers are attached to the molecules and photographed (d). The last phase is the base calling process in which the imaged colors of each spot are translated into sequences of bases (e).

analysed must be recovered from these short reads either through alignment or through assembly. Errors within reads, repetitive regions of the genome, single nucleotide polymorphisms and complexity of genomic sequences make this task very challenging.

1.3.1 Platforms for next generation sequencing

Each proprietary NGS platform has its own technological procedure. The most common platforms used are the Genome Sequencer FLX by 454 Life Sciences and Roche, Genome Analyzer by Illumina and SOLiD developed by Applied Biosystems. Even though all these procedures are different, they can be described in terms of three separate phases (Metzker, 2010). An outline of the sequencing process is depicted in Figure 1.1.

The first phase entails preparing a sequence library, in which the DNA sequence of interest must be shredded into shorter sequences of roughly equal size. These sequences have to be attached to a specific carrier, enabling parallel sequencing of a thousand to million molecules. Preparing the library may also include sequence amplifications, where the molecules are amplified to increase signal strength in later procedures.

	read length (*bp*)	throughput (reads)
GS FLX	700	1M
HiSeq 2000	100	6G
SOLiDv4	50	1.4G

Table 1.1: **Comparison of read length and maximal throughput of most common sequencing platforms.** Data reported by Liu et al. (2012).

In the second phase the molecules are sequenced using chemical reactions. Sequencing usually entails multiple cycles of adding fluorescent markers and imaging resulting molecules. In each cycle a complementary nucleotide carrying a marker is attached, the molecules are photographed and the marker is removed. Nucleotides are distinguished based on the observed colour spectrum. Modern platforms are able to sequence only a small number of nucleotides resulting in very short sequences ranging from 25*bp* to 700*bp* in length, depending on the sequencing technology.

The last phase is the data analysis. Resulting colour spectra have to be translated into sequences of bases which can be used for further investigation. Produced short reads have to be either aligned to an existing reference genome or assembled *de-novo*. Further steps are dependent on the type of experiment being carried out.

The main interest of this report are the methods for analysing data obtained in the last phase of NGS experiment. However, it is very important to appreciate the whole process of generating a dataset. Understanding the sequencing procedure and its limitations and possible errors enables us to create methods which make use of this knowledge and produce results of higher accuracy.

Pyrosequencing

Technology provided by 454 Life Sciences/Roche was the first widely available, next generation sequencing platform (Mardis, 2008). It uses pyrosequencing to determine nucleotide sequences.

The library preparation is based on emulsion polymerase chain reaction (PCR) (Shendure and Ji, 2008). Molecules are broken into smaller sizes and attached to beads. The beads with molecules are inserted into droplets in which the molecules can be amplified via PCR. After purification the molecules are deposited into Picotiter Plate (PTP) wells in which the main chemical process takes place.

The second phase consists of addition of beads with enzymes to the PTP wells. Afterwards a nucleotide solution causing bioluminescence is applied to the PTP and the light emission caused by chemical reaction is recorder by CCD camera. For this approach it is difficult to distinguish longer continuous regions of same base which can cause errors in resulting sequence (Mardis, 2008).

Genome Sequencer FLX by 454/Roche is able to produces the longest reads out of the Second generation platforms, see Table 1.1. However, the number of reads sequenced is in the order of hundreds of thousands and cost of sequencing per base pair is the highest out of currently applied NGS technologies (Wall et al., 2009).

Sequencing via reversible termination

Sequencing using cyclic reversible termination was introduced by Solexa in Genome Analyzer 1G, which was later acquired by Illumina. In this case the sequences are attached to a glass slide, also called a flow cell, on which they are amplified by the use of bridge PCR (Metzker, 2010). Separated clusters of molecules are formed on the flow cell, which are ready to be sequenced.

The sequencing proceeds in cycles in which universal primers are used to attach a complementary nucleotide with fluorescent dye to every molecule. The addition of a single nucleotide is achieved by termination of DNA synthesis. After the imaging step determines the nucleotide using total internal reflection fluorescence, cleavage removes the 3' blocking group. The number of such cycles is dependent on desired read length. A base caller determines the base of each nucleotide and estimates quality of the call. The most probable error for this kind of procedure is a base substitution (Shendure and Ji, 2008).

The first generation Genome Analyzer was capable of producing tags of length around $25bp$ to $36bp$. The read length of current generation devices, HiSeq 2000, increased substantially up to $150bp$ reads. The number of generated reads is now of the order of billions of short sequences per run.

Sequencing using ligation

Support oligonucleotide ligation detection (SOLiD) is a sequencing platform developed by Applied Biosystems (AB). The preparation is based on emulsion PCR similar to Genome Sequencer FLX, with the exceptions that in SOLiD the beads are attached to a glass plate (Metzker, 2010).

The sequencing process makes use of DNA ligation instead of PCR. The DNA ligase is used to attach a universal complementary primer. After observation of fluorescent marker, cleavage removes the ligase with dye. This process can be applied to every fifth position in the sequence. The whole cycle is restarted by removing the extended primer and starting the whole process shifted by one nucleotide.

The method uses two base probes which provide information about two neighbouring bases, thus each base is encoded twice providing less error prone results. SOLiD produces specific colour space encoding of sequences which can be either translated into regular base-pair encoding or it can be used for higher precision alignment.

While the reads produced by this platform are usually $50bp$ long, the number of reads is close to that of HiSeq devices and the base error rate is lower (Liu et al., 2012).

Other methods

More recently introduced sequencing devices are HeliScope by Helicos Bioscience, Personal Genome Machine by Ion Torrent and RS by Pacific Biosciences.

The HeliScope uses a single molecule sequencing approach with high sensitivity fluorescence instead of amplification of molecules. It avoids the amplification step which can introduce errors and biases (Metzker, 2010).

The Personal Genome Machine employs a novel approach, using a semiconductor detector instead of fluorescent imaging. It detects the changes in pH due to proton release during synthesis (Quail et al., 2012). As the name suggests, it is aimed to be a *benchtop* device with lower acquiring cost while providing a novel way of sequencing approach.

In 2010, Pacific Bioscience introduced its single-molecule real-time sequencing device RS. It is regarded as Third Generation Sequencing due to the avoidance of PCR during the preparation and sequencing of molecules in real time. The RS is able to read long continuous sequences of molecules resulting in variable length, with mean read length up to $2566bp$ (Liu et al., 2012). While being a promising future technology, at the moment, the error rate is higher and sequencing throughput is lower than NGS devices.

Other promising technology suitable for the third generation sequencing is nanopore sequencing where a molecule driven through a suitable nanopore would

change the ionic current through the nanopore (Branton et al., 2008). The characteristic change in ionic current can then be used to identify individual bases. However, this kind of device has not been released yet and the technology is still under development.

An intriguing alternative approach to NGS is the G.007 Polonator using sequencing by ligation similar to the SOLiD devices (Shendure and Ji, 2008). It was aimed to be developed as an open source platform providing low cost access to sequencing. Unfortunately it falls behind the commercial devices in terms of the read length and throughput (Metzker, 2010).

1.3.2 Applications of high-throughput sequencing

The technology of high-throughput sequencing was originally designed to study and explore genomic sequence. Except for the improved throughput and reduced cost, there are two properties shared by the NGS technologies that enabled its use in other areas. Firstly, as long as the sample provided consists of fragments of DNA molecules, the origin of the molecules does not matter. This, for example enables analysis of RNA through transformation of RNA into complementary DNA (cDNA). Secondly, the molecules present in the sample are fragmented and sampled almost uniformly, enabling their quantification by read count. Note that while biases have been reported in the sequencing output, these biases tend to be systematic and thus can be accounted for with a careful analysis, see Section 1.5.2.

Here we present an example of different applications of NGS technologies applied to genomic, proteomic and transcriptomic problems. These are the most common applications of NGS and hopefully provide an overview of the spectrum of problems that can be addressed by this technology. Many other variations of these approaches and combinations of high-throughput sequencing with other experimental methods have been reported.

De-novo discovery

De-novo discovery is the basic way to study unknown genomic sequences. Apart from well studied model organisms, e.g. human, domestic mouse or *C. elegans*, the majority of species have unknown genome. In de-novo sequencing, the unknown

genome is sequenced by the high-throughput sequencing technology. The resulting reads then have to be assembled into consecutive chromosomal sequences. This can be done by looking for long-enough overlaps of reads.

Assembling the short reads generated by NGS in an efficient way while accounting for errors occurring in the sequencing output requires novel algorithmic approaches. While many methods still rely on the longer reads produced by Sanger sequencing, new techniques that successfully use high-throughput sequencing for this task have been presented (Zerbino and Birney, 2008; Simpson et al., 2009; Miller et al., 2010; Li et al., 2010c; Simpson and Durbin, 2012).

For de-novo discovery it is preferable to use technology that produces long reads as it enables longer overlaps of reads and disambiguates short repetitive regions. The sample might have to be sequenced multiple times in order to produces sufficient read coverage as high sequencing depth is also necessary. While the majority of genomic sequence can be assembled through the use of NGS, long, repetitive regions and duplicated sequence are limitations of this approach (Alkan et al., 2011). A comparison and evaluation of some of the currently used applications for de-novo assembly of NGS data was done by Salzberg et al. (2012).

Analysis of genetic variation

Even for known genomes, studying variations between individuals is important. For example, most human genomic sequence is shared by every individual and only around 2% bases vary [1]. Changes at these bases are referred to as Single Nucleotide Polymorphisms (SNPs) and are the fundamental way individuals within one species differ.

The variations of genomic code and their relation to traits are highly researched topics. Apart from SNPs, other variation such as sequence insertions and deletions (indels) are also important. While some variants are known to be directly responsible for genetic disorders, others might have indirect relation to important changes in phenotype.

NGS can be directly applied to the SNP and indel discovery problem in known genomes. Given a reference genome of the sample being sequenced, reads generated by the high-throughput sequencing are firstly aligned to the reference. The

[1] Build #138 of the database of genetic variations, dbSNP (Sherry, 2001), lists 62.7M variations.

alignment can be described as the simple problem of finding positions of substrings within long reference string made much harder by sequencing errors and the variations themselves. We describe the alignment process in greater detail in Section 1.4.4. Once the reads are aligned to the reference, base variations supported by multiple reads are selected as SNP candidates. Nielsen et al. (2011) reviews common software packages available for SNP detection from aligned NGS reads.

Unlike the case of *de-novo* assembly, the length of NGS reads does not play an important role for SNP discovery as the alignment is a much easier task. Nevertheless, high sequencing coverage is important for discriminating between sequencing errors and true variations. Low error rate, especially in terms of base substitution, is also desired, even though it can be substituted by increased depth of sequencing.

Protein — DNA interactions discovery

Chromatin Immunoprecipitation followed by sequencing (ChIP-seq) is a method for exploring protein — DNA interactions through discovering the binding sites of proteins, such as transcription factors and histones. A protein of interest is mixed with chromatin treated by a chemical reagent, enabling formation of cross-links with binding sites. Chromatin is then sonicated into smaller fragments, followed by the immunoprecipitation, a process of adding a protein specific antibody to isolate protein with bound DNA. After purification of immunoprecipitated chromatin, the cross-links are reversed separating the protein from DNA, which is quantified through the use of sequencing.

The immunoprecipitated chromatin is sequenced with a high-throughput sequencing device, producing reads that have to be aligned to the reference. Because the isolation in ChIP is not complete, the method only enables enrichment of the binding sites instead of direct selection. As the reads align to the entire genome, the binding sites have to be identified through searching for enriched regions, also referred to as peaks. Detecting peaks that correspond to true binding sites is the most difficult part of the analysis.

For a review of the ChIP-seq methodology, its advancements and caveats please refer to one of (Park, 2009; Pepke et al., 2009; Hoffman and Jones, 2009). More detailed evaluation of accuracy and precision of previously published algorithms can be found in (Laajala et al., 2009; Wilbanks and Facciotti, 2010).

Sequencing of RNA

While all previous applications were primarily targeted at the analysis of genomic DNA, high-throughput sequencing can be also used to study molecules of RNA. RNA molecules present in cells can be extracted and reverse-transcribed into complementary DNA (cDNA), which is then sequenced and analysed by high-throughput sequencing technology. The application of NGS technologies for sequencing RNA is commonly known as RNA-seq and is described in detail in Section 1.4.

1.4 RNA-seq

1.4.1 Transcriptome

Information encoded in the DNA defines all living organisms. The most important part of the information is stored in the form of genes, which are sub-sequences of the DNA serving as source code for all other molecules being created. The information stored in genes is copied, or transcribed, to molecules of RNA by RNA polymerase enzymes. While some of the molecules serve regulatory function and others serve as a basis for building proteins, the sum of all RNA present in the cell is the transcriptome.

As the genetic code is constant, the differentiation of cells and tissues is done via regulating the use of the genetic code. This is done on several levels, one being the regulation of transcription. Analysis of the transcriptome provides a key step in the exploration of regulatory functions of living organisms, deciphering of gene functions and detection of genetic disorders.

The transcriptome consists of different types of RNA molecules: messenger RNA (mRNA), ribosomal RNA (rRNA), transfer RNA (tRNA), micro RNA (miRNA) and other 'non-coding' RNA. The mRNA is the only RNA that is translated into proteins which are the main building blocks of living organisms, hence is referred to as the coding RNA and its genes are called coding genes. While most studies of the transcriptome focus on the mRNA as it is the precursor for proteins, analysis of non-coding RNA is important for understanding many other cellular mechanisms.

Different transcripts of the same gene are often referred to as isoforms of the gene. Despite being from the same genetic locus, their function can vary. A small

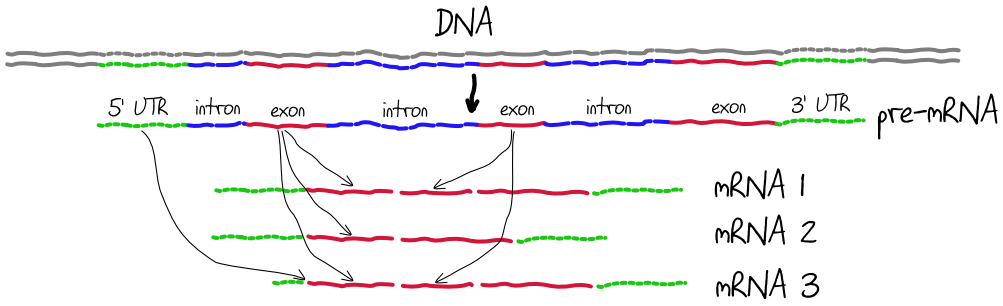


Figure 1.2: **Illustration of transcription.** The gene encoded on a specific strand of the DNA is transcribed into precursor RNA (pre-mRNA). The final transcripts are formed by a splicing mechanism which removes introns from the sequence. The splicing mechanism can also remove parts of coding sequences or entire exons, resulting in multiple transcripts of a single gene.

change in a mRNA can lead to a protein with different function being translated.

Most genes, especially the coding genes, consist of smaller units: exons and introns. Exons carry important information about the gene's product while the function of introns has not yet been fully explained. Transcription of genes into RNA has two functionally separate steps which can happen concurrently. First the gene is copied into precursor RNA (pre-mRNA) [2] which is then spliced into the actual transcript of RNA. The process is outlined in Figure 1.2. During the process of splicing, all the introns and in many cases also part of the exons are spliced out of the original sequence. A gene can be spliced in more than one way, thus leading to the creation of multiple transcripts — different RNAs originating from a single gene.

1.4.2 RNA-seq for expression level quantification

We say that a gene is expressed when it is being transcribed and we refer to the abundance of its transcripts within a sample by the term expression level. The gene expression level can be used as a proxy measurement of its activity despite the fact that the abundance of transcripts does not imply their actual use. Nevertheless, high abundance of transcripts signifies some increased activity, while no transcript molecules directly imply gene's idleness.

The expression of genes in cells and tissues have been of great interest to

[2] The abbreviation pre-mRNA is used for referring to all primary transcripts, not just those of mRNA as the name might suggest.

scientists in various areas. Thanks to the properties of NGS technologies, sequencing the transcriptome can be directly used to measure the abundance of RNA molecules. Unlike previous methods, NGS produces reads along the entire length of transcripts and thus can be used to detect gene isoforms. This enables the use of RNA-seq to measure expression levels of transcripts.

The main premise of transcript expression level quantification based RNA-seq is following. Under ideal conditions the expected number of reads, $E[C_m]$, that are sequenced from a transcript m, is directly proportional to the number of fragments of that transcript within a sample.

$$E[C_m] = D \cdot (F \cdot \theta_m), \tag{1.1}$$

where D is a constant representing sequencing depth, F is the total number of fragments and θ_m is the relative proportion of fragments of transcript m. The number of fragments of a transcript, $(F \cdot \theta_m)$, is directly proportional to the number of molecules multiplied by its effective length. Here the effective length expresses the number of different fragments that can be generated from a transcript, which can be calculated as $l_m^{(eff)} = l_m - l_f + 1$, where l_m is the length of the transcript and l_f is the length of a fragment. So under ideal conditions, assuming a constant fragment length, we can express the expected number of reads of a transcript in terms of abundance and length as follows

$$\begin{aligned} (F \cdot \theta_m) &= K \cdot l_m^{(eff)} \cdot (\text{abundance}_m), \\ E[C_m] &= D \cdot K \cdot l_m^{(eff)} \cdot (\text{abundance}_m), \end{aligned} \tag{1.2}$$

where K is a constant that scales the number of produced fragments and abundance_m denotes the abundance of molecules of transcript m within a sample. Given the number of reads sequenced from a transcript, we can use Equation 1.2 to calculate the transcript abundance. Note that the constant factor $D \cdot K$ is unknown and thus it is impossible to quantify absolute abundances using RNA-seq. We can either estimate proportional abundances or abundances under specific sequencing output.

The ideal conditions in this case refer to a process in which fragments are sampled uniformly along transcripts and reads are sequenced uniformly from all fragments. While the ideal conditions are not practically achievable, the great amount of sequenced reads enable accurate approximation of abundance through

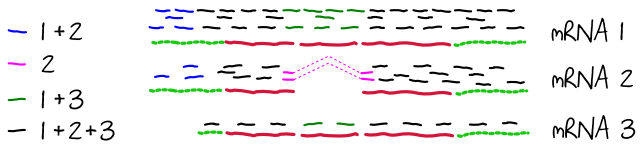


Figure 1.3: **Example of the ambiguity of reads' alignments caused by transcripts sharing multiple exons.** Reads above a transcript were sequenced form that transcript. The colour denotes to which group of transcript would reads align. Most of the reads (black) align to all three transcripts, blue reads align to the first and second transcripts, purple reads align only to the second transcript and green reads align to the first and third transcripts.

assuming the equality given in Equation 1.2. Furthermore, techniques for correcting systematic biases caused by the fragmentation and sequencing processes have been proposed and are discussed in more detail in Section 1.5.2.

Transcript ambiguity

The main difficulty of quantifying transcript expression levels is that the exact origin of reads is lost. While the majority of the sequenced reads can be aligned to a unique position within the genome, this does not determine the actual transcript of origin. Reads mapping to an exon could have originated from any transcript that contains that exon. Therefore the number of reads sequenced from a transcript cannot be learned directly from the sequenced reads due to transcript ambiguity.

We illustrate the problem in Figure 1.3 using three transcripts of a single gene. Multiple exons are shared between the transcripts leading to many reads with ambiguous origin. This ambiguity is in most cases either resolved based on other reads providing extra evidence of the transcript abundance, or modeled within a probabilistic framework that considers the joint distribution of the data. Nevertheless, there are cases for which exact quantification is impossible. Lacroix et al. (2008) proved that even under the ideal conditions, with sufficient coverage, some transcripts cannot be distinguished using just the RNA-seq reads.

In this thesis we address the problem of quantifying transcript expression using Bayesian probabilistic modelling. We propose a model over all the observed reads and their alignments. Using Bayesian inference methods, we can infer the posterior distribution over transcript expression. Providing a full posterior distribution instead of just point estimate provides information about the uncertainty of the estimated expression level. Transcripts that are difficult, or even impossible, to

quantify exactly will be associated with a posterior distribution of expression level with high variance. On the other hand, transcripts from genes with a single transcript or multiple transcripts with high coverage will have lower variance in their posterior estimate. Our transcript expression quantification approach and its results are presented in detail in Chapter 2.

Gene expression levels

In many cases the abundance of all gene products is of interest. We and others (Trapnell et al., 2013) have proposed that the best way to quantify gene expression is by estimating transcript expression first and then add up abundances of appropriate transcripts to produce gene expression.

Many reports estimate gene expression using RNA-seq reads directly, without transcript-level quantification. There are two basic approaches that can be used for counting reads that map to a certain gene. One uses the *union* method and adds up reads of all constitutive exons. In this case the effective length of the counting region is a mixture of the effective lengths of the underlying transcripts,

$$l_g^{(eff)} \propto \sum_{m \in g} \theta_m \cdot l_m^{(eff)}. \tag{1.3}$$

The union approach is suitable for genes with a single transcript or when it is known that the gene is being spliced in the same way throughout the analysis. However, once the gene can be spliced into transcripts of varying length, this measure is inconsistent as the effective length of the counting region changes.

Wang et al. (2010c) compared the union approach of gene expression estimation with adding up isoform expression. They showed that estimating isoform expression levels and adding those together leads to better estimates of gene expression with lower uncertainty.

The alternative is to use the *intersect* approach in which only reads mapping to exons shared by all gene's transcripts are being counted. Thus the effective length is a mixture of proportional effective lengths of the *intersect sequence* within transcripts,

$$l_g^{(eff)} \propto \sum_{m \in g} \theta_m \cdot l_{(i,m)}^{(eff)}. \tag{1.4}$$

Assuming uniform read coverage, the effective lengths of the intersect sequence within transcripts are equal, hence the method provides unbiased measure of the

abundance of gene products. However, for genes with many different transcripts the intersect sequence can be relatively short, leading to the majority of data being discarded. In such cases, abundance estimates for low-covered genes or long genes with short intersect sequence can be adversely affected.

Another problem arises with biased distributions of reads along transcripts, when the effective lengths of the sequence can vary depending on transcript and surrounding exons.

Measures of expression levels

The most basic measure for RNA-seq gene expression is the *read count* measure (C), where each gene or transcript is characterized by the number of reads that align to that particular gene. While this is a natural way of looking at the result of sequencing process, it is not optimal for comparison of gene expression as read count is proportional to sequence length as was discussed above.

A second difficulty with expression level measured in read counts arises in comparisons of independently sequenced samples. Two samples sequenced to a different depth will have a different total number of reads reported hence varying counts per gene. This can be avoided by adjusting the counts so that the total number of reads is the same, or by applying a more advanced normalisation method, for details see Section 3.3.

Mortazavi et al. (2008) introduced the *RPKM* measure or the 'Reads Per Kilobase of length per Million reads sampled'. This metric adjusts for varying sequencing depth and gene length in order to make the expression of genes comparable within and between samples. It is straightforward to compare expression levels of two genes within one sample as well as to compare the same gene represented by various transcripts between two conditions.

The calculation of a gene's RPKM is a simple addition of RPKM of its transcripts. The RPKM of each transcript has already been adjusted by the effective length of the transcript, hence the sum of each transcripts' RPKM is a consistent measure of the abundance of gene products despite alternative splicing.

Trapnell et al. (2010) use the term FPKM, or 'fragments per kilobase of length per million reads sampled', to distinguish analysis containing paired-end reads, in which pairs of reads are sequenced from both ends of one fragment.

Another form of normalized proportional measure is the relative abundance

of transcript fragments, here denoted by θ, which can be expressed as

$$\theta_m = \frac{C_m}{N}, \tag{1.5}$$

for a read count of transcript m, and total read count $N = \sum_{m=1}^{M} C_m$. While this is not a suitable measure for genes due to the length ambiguity, it can be used for transcripts as it characterizes each transcript by the proportion of its fragments within the original sample. This is a natural way of looking at the expression from a generative view used in a generative probabilistic model.

Given a transcript m and its effective length $l_m^{(eff)}$ conversion between the measures is straightforward,

$$RPKM_m = \frac{C_m}{l_m^{(eff)}/10^3 \cdot N/10^6} = \frac{\theta_m}{l_m^{(eff)}} \cdot 10^9. \tag{1.6}$$

Differential expression analysis

Comparing expression estimates from different samples is an important stage of many research studies. Whether the samples are from different tissues, treatments or time points, the task is always the same, to find genes that exhibit different expression patterns between conditions. We refer to this task as the Differential Expression (DE) analysis.

There are various levels at which samples from different conditions can be compared. One can compare the abundance of all gene products, study alternative splicing patterns between conditions or look at the abundances of transcripts. In each of the settings, the problem is to identify significant differences between the conditions.

The matter of significance is important when performing either kind of DE analysis (Auer and Doerge, 2010; Fang and Cui, 2011). It is essential that DE analysis is able to distinguish significant changes of expression between conditions from the fluctuations of expression within one condition. There are two parts of the expression level fluctuations, the technical variance and biological variance.

The technical variance usually refers to the measurement error due to the technology being used. In the case of sequencing it is the effect of random sampling of reads as well as the limitations of expression estimation method due to insufficient coverage or transcript ambiguity mentioned above. The technical variance can be overcome by improving the technology being used, for example

by increasing the coverage through higher depth or multiple sequencing runs, or by increasing the read length which can help distinguish ambiguous transcripts.

The second source of fluctuations is the biological variance of expression levels caused by natural changes of the abundances of gene products which fluctuates within cells and tissues. This variability is independent of the technology being used and cannot be avoided. It can only be accounted for when deciding whether the observed difference is significant or not.

The extent of biological variance depends on the gene or transcript of interest (Anders and Huber, 2010). While some genes have stable expression levels within a condition, other genes might have significant variations of expression levels within condition. The biological variance can be assessed by performing the analysis on multiple replicates of each condition, also referred to as biological replication. Biological replication has been previously used in DE analysis based on other technologies such as microarrays (Dudoit et al., 2002) and it is essential for RNA-seq based DE analysis as well (Fang and Cui, 2011). Through the analysis of biological replicates, it is possible to determine the extent of the biological variation of individual transcripts and account for these fluctuations when determining the significance of changes between conditions.

We propose a novel approach for the DE analysis using Bayesian inference that can be used for detecting changes in both transcript and gene expression levels between multiple conditions. The method, presented in Chapter 3 uses results from our expression estimation procedure which includes estimate of the technical variance of each transcript. The technical variance within each sample is combined with results from multiple biological replicates to assess the significance of changes in expression level between conditions.

Alternative ways of expression level quantification

The most common technology used for gene expression analysis before NGS are microarrays. A microarray is a *lab on chip* type of assay that uses a 2D matrix consisting of tens of thousands of probes, which are designed to bind to specific molecules. After the sample is put on to the array, the array is scanned in order to determine intensities at each probe. The intensities correspond to the amount of bound molecules and thus can be used to measure the abundance of the molecules within a sample.

Microarray technology provides a widespread approach for gene expression

studies albeit having many drawbacks. The nature of microarray design limits its use and properties of results. The probes have to be prepared to match the molecules of interest, prohibiting the discovery of new transcripts and genes. The microarray probes bind only with a specific sequence tag of genes and thus there is no way of distinguishing between different transcripts binding to the same tag. Another disadvantage of this technology is that the signal is produced via imaging of fluorescent dyes. This produces a continuous signal that is harder to compare between samples and can be easily saturated. While microarrays can have up to tens of thousands probes on each array, measuring expression level of transcripts of organisms with more than one hundred thousand transcripts would require multiple arrays to be used.

To eliminate some of the disadvantages of microarray, methods such as Serial Analysis of Gene Expression (SAGE), Cap Analysis Gene Expression (CAGE) and Massively Parallel Signature Sequencing (MPSS) using Sanger sequencing were developed (Wang et al., 2009). The main principle of these methods is to assay only one short sequence, or tag, from each molecule. Instead of sequencing the whole gene, only the tags are being sequenced, thus reducing the cost of the sequencing. In contrary to microarrays, SAGE and similar methods avoid problems with saturation and require no molecule-specific probes. However, it is still not possible to study expression of individual gene transcripts with this type of methods as many transcripts can share the same tag.

1.4.3 RNA-seq protocol

All of the current NGS devices are capable of sequencing molecules of DNA. To enable the analysis of the transcriptome, RNA molecules have to be reverse-transcribed into cDNA which can be analysed by a sequencing device. Here we provide an overview of the preparation process of RNA molecules before the actual sequencing. Understanding this process is important for RNA-seq data analysis and expression level quantification as it can create biases that are only specific to sequencing of RNA (Hansen et al., 2010).

(1) Selection

An RNA-seq experiment begins with the extraction of all RNA from a studied tissue. This could be RNA from the nucleus, cytosol or from the entire cell. Most

RNA-seq experiments focus on the protein coding genes, hence follow with the isolation of mRNA. The mRNA molecules are polyadenylated, i.e. the 3' end is followed by a sequence of Adenine nucleotides, also referred to as poly-A tails. The tails are used to extract the molecules with complementary poly-T sequence attached to plates or magnetic beads (Wang et al., 2009). This, however, extracts also the long non-coding RNA which has the poly-A tail as well.

Other methods have been developed for studying a broader spectrum of RNA sequences. Instead of selecting molecules with poly-A tails it is possible to remove ribosomal sequences which form over 90% of RNA within cell (Wilhelm and Landry, 2009). Ribosomal RNA (rRNA) contains highly conserved sub-sequences which can be used to separate rRNA using beads with complementary nucleotide sequences. Removal of the rRNA increases the proportion of mRNA and non coding RNA within the sample and enables their sequencing with sufficient depth.

(2) Reverse transcription and fragmentation

The molecules of RNA are converted into cDNA through the means of reverse transcription (RT) and fragmented. The first strand synthesis of RT has to be initiated by a primer attached to the RNA. In the case of mRNA poly-T primer sequence can be used to bind to the poly-A tail. Otherwise random primers, which bind anywhere along RNA, can be used. Subsequently second strand synthesis is used to create the complementary second strand of cDNA.

As most current technologies only sequence relatively short ends of molecules, the transcripts are typically sheared into smaller fragments. With the use of random primers, it is possible to fragment the RNA molecule before RT. In cases where this is not desired or poly-T primer is being used, fragmentation is applied to the cDNA molecules after RT.

The choice of a fragmentation methodology and primer affects the distribution of reads. The use of poly-T primer can lead to bias towards 3' end of a transcript due to incomplete first strand synthesis (Wilhelm and Landry, 2009). RNA fragmentation on the other hand causes under-representation of both ends of transcripts (Wang et al., 2009). Random primers can similarly cause bias towards reads originating from certain positions (Hansen et al., 2010).

Reverse transcription transforms a single strand of RNA into double stranded cDNA molecules with one strand being equivalent of the RNA and the other being reverse complement. In a standard protocol this process loses the strand

association of the original molecules (Wilhelm and Landry, 2009). Alternative preparation protocols which preserve the strandedness can be applied when the information is needed for further analysis. Review and comparison of preparation protocols that preserve the strand information can be found in Levin et al. (2010).

(3) Addition of adaptors and primers

To enable processing of the sample by a sequencing device, platform specific adaptors and sequencing primers have to be attached to the cDNA molecules. The adaptors are used to enable attachment of molecules to a support and to enable clonal amplification, while primers are used for initiation of the sequencing process.

Most current devices enable attachment of two sequencing primers to each fragment and sequencing each of its ends. This is so called pair-end sequencing as the reads are reported in the form of paired tags or *mates*. While the tags can be treated independently, each as a separate read, using the pairing information is useful for downstream analysis, especially isoform deconvolution and quantification. For this reason we tend to refer to the paired tags as to one *paired read* and always treat them jointly.

As the sequencing primers are attached to each strand of cDNA, the two mates of a paired read are sequenced one from each strand. This means that while one mate will align to the transcript sequence, the other read will have complementary sequence. Also, due to the sequencing reaction starting from the primers attached to fragment's ends, one read is reversed in terms of transcript orientation.

For strand specific protocols, the paired reads will always align to the reference transcripts in the same order. For example in dUTP protocol this is achieved by firstly sequencing the strand which was generated by first strand synthesis. With respect to the transcript sequence the first mate will always be reverse complement located downstream of the second mate, while the second mate will align concordantly with transcript.

(4) Sequencing and base calling

After attachment of primers and adaptors, the cDNA can be processed by the sequencing device. The actual process of sequencing for the three most popular platforms is outlined in Section 1.3.1. Except for the Ion Torrent, the NGS devices

obtain actual sequence of each read using imaging of fluorescent markers attached to each nucleotide, a process which is also called base calling.

In addition to the base calls for each nucleotide, modern sequencing platforms can also estimate the likelihood of error for each call. These are reported as quality scores, usually in form of *Phred* format.

The resulting output of each RNA-seq experiment is then a number of reads, usually of the order of millions. Each read is represented by a sequence of bases and a sequence of quality scores of the same length. In the case of paired-end sequencing, the tags from one fragment are reported as a pair of two reads.

1.4.4 Read alignment

Alignment of reads is the process of searching for positions within a reference genome or transcriptome where the read sequence aligns, or matches, with the reference sequence. For most subsequent analyses, the reads generated by an RNA-seq experiment have to be aligned. Only in cases when the reference, in the form of a transcriptome or a genome, is unknown or incomplete do the reads have to be assembled instead.

Alignment, also called mapping, produces a set of alignments for each read. These are positions where the read matches the reference perfectly or with some number of differences such as mismatches between individual bases. The differences are usually caused by sequencing errors or by variation of the sequencing sample from the reference, i.e. SNP or insertion or deletion. In very rare cases can a read be mapped to an incorrect location with only few mismatches. Most reads can be matched uniquely to a single position in the reference sequence. The mapping position determines the sequence and location which was sequenced for that particular read.

Genomic alignment

The adoption of NGS methods required development of new approaches for read alignment. Methods designed for processing reads for Sanger sequencing, such as BLAT (Kent, 2002), were not designed to work with the type of data produced by NGS devices. The sequences are very short in comparison with the reference, they usually contain base mismatches and other variations and most importantly, the amount of sequences is orders of magnitude higher.

All current aligners use indexing in order to speed-up the alignment process. Indexing the sequences allows fast retrieval of candidate positions which can then be evaluated more closely for each read. While some aligners, such as MAQ (Li et al., 2008), index the reads that are to be aligned, most aligners pre-index the reference sequence. The former strategy can be more effective as more time can be spent on indexing which is done once per reference.

There are two main types of indexing used at the moment. One group of algorithms uses hashing based indexes, where k-mers or seeds are indexed in a hash table. The following algorithms use hashing indexes: BFAST, GASST, NextGenMap, PerM, SOAPv1, SHRiMP 2, Stampy (Homer et al., 2009; Rizk and Lavenier, 2010; Sedlazeck et al., 2013; Chen et al., 2009; Li et al., 2008; David et al., 2011; Lunter and Goodson, 2011).

The second approach uses suffix tries combined with FM-indexes (Ferragina and Manzini, 2000) and the Burrows-Wheeler transform (Burrows and Wheeler, 1994) for searching within the index. Example algorithms in this group are Bowtie, Bowtie 2, BWA, SOAP2 (Langmead et al., 2009; Langmead and Salzberg, 2012; Li and Durbin, 2009; Li et al., 2009b).

A detailed methodology review of NGS alignment methods was written by Li and Homer (2010). Each method has its own advantages and drawbacks and with so many available it can be difficult to choose the correct tool. Most users are usually concerned with the performance of the aligner and its run time requirements. Ruffalo et al. (2011) provide a comparison of six popular alignment tools and look at the performance dependence on base errors and indels. Another comparison of thirteen popular aligners was conducted by Lindner and Friedel (2012), who also try to estimate the optimal set of parameters for each method and provide more detailed performance measure in terms of precision, recall and F-measure.

Splice-aware alignment

All of the above mentioned aligners are designed for exact alignment of reads to the genome while allowing for a certain number of variations. This kind of approach works well for genomic NGS reads and a subset of reads from RNA-seq experiment. However, as RNA-seq produces short reads from the transcriptome, reads that span splice junctions will not align with the genomic reference.

Splice aligners are designed specifically for mapping transcriptomic reads to

genomic reference, taking into account splice junctions. On the one hand, there are splice aligners such as GSNAP or STAR (Wu and Nacu, 2010; Dobin et al., 2013), which use own alignment algorithms that handle junction reads directly. On the other hand, some splice aligners build upon one of the previously mentioned genomic aligners while providing extra functionality that handles the junctions. MapSplice developed by Wang et al. (2010a) divides reads into shorter sub parts and uses Bowtie to align them to the reference. Even if one of the sub parts is from a splice junction and cannot be aligned, other sub parts will be aligned and can be extended up to a exon boundary. A similar approach was taken by Au et al. (2010) in SpliceMap, where each read is divided into $25bp$ parts and each part is mapped individually using Bowtie or another genomic aligner.

An alternative approach was presented in QPALMA (De Bona et al., 2008), which uses vmatch (Abouelhoda et al., 2002) to align exonic reads, then uses these reads to predict exon sequences. Based on the exonic sequences and known splice junctions, the algorithm predicts new splice junctions which are used for mapping of the rest of the reads. TopHat (Trapnell et al., 2009; Kim et al., 2013) improves this approach by using Bowtie as a more efficient alignment algorithm and omitting the predictive step. Instead of predicting the splice junctions, all possible junctions are assembled and used for alignment of reads. This process can be further simplified if annotation of exon boundaries does exist.

Grant et al. (2011) present an evaluation framework and splice read generator for comparing splice alignment algorithms. They use this framework for comparison of previously published algorithms and evaluation of a newly proposed approach, RUM. RUM exploits the speed of Bowtie for initial alignment to the genome and transcriptome and subsequently aligns unmapped reads using BLAT. It performs similarly to the best current approaches with relatively low run-time complexity.

Transcriptome alignment

Many RNA-seq experiments involve organisms with a known reference genome, which is well annotated in terms of gene locations, exon boundaries and transcript isoforms. In these cases, it is usually much more convenient to align reads to the transcriptome sequence directly.

Aligning RNA-seq reads to the transcriptome simplifies alignment by avoiding the necessity of handling junction reads. Moreover, the transcriptome is an order

of magnitude smaller than the genome, making the alignment much faster.

The transcriptome reference can be download as a cDNA sequence from a repository provided by Ensembl (Flicek et al., 2013), or constructed by the Table Browser maintained by UCSC (Karolchik et al., 2004; Kuhn et al., 2013). This reference represents sequences of cDNA as they are sequenced, which avoids the entire complication with spliced alignments. Genomic aligners can be used directly to align all the reads. Only a fraction of reads, which may come from yet un-annotated splice junctions, novel genes or intronic sequences will fail to align. The number of such reads depends on the completeness of the annotation and selection protocol.

There is a caveat that has to be considered when using genomic alignment algorithms to align RNA-seq reads directly to the transcriptome. As opposed to junction reads that were difficult to align to the genome, now exonic reads will have multiple alignments. Most exons and also some splice junctions can be shared by multiple transcript isoforms of the same gene and in such cases reads from these sections will align equally to all those isoforms. Downstream methods that make use of the alignments, such as RSEM (Li et al., 2010a) or the method presented in Chapter 2, require all these alignments. Not all genomic aligners have the option of reporting multiple alignments per read, some aligners discard ambiguously mapped reads while some report only alignments with the best alignment score. With the correct optional setting, aligners such as Bowtie, Bowtie 2, SOAP 2 and SHRiMP can report multiple alignments per read.

In some cases, the transcriptome alignments are required by downstream analysis, but the annotation might be incomplete or nonexistent. Then it is possible to use a splice-aware alignment combined with a program for transcript assembly to create a new transcriptome and subsequently align to the newly assembled reference.

Overall there are numerous choices available in terms of alignment algorithm. Fonseca et al. (2012) presented an overview of both genomic aligners and splice-aware alignment algorithms, which is being updated at `http://wwwdev.ebi.ac.uk/fg/hts_mappers/`. While performance evaluation of more than 60 aligners would be impossible, the authors provide a useful comparison in terms of capabilities and features. As an example, narrowing the choice of mappers that would be useful for analysis presented in this report can be much easier by selecting aligners that can output multiple alignments per read, align paired-end reads

and provide output in the SAM format.

1.5 Related work

1.5.1 Initial evaluations of RNA-seq technology

The first studies involving NGS of RNA assessed the possibilities of using NGS for analysis and exploration of the transcriptome. Weber et al. (2007) used Pyrosequencing to study the transcriptome of Arabidopsis. The authors report very deep coverage of genes by the 541852 sequenced tags in two runs. Moreover, they report reads mapping to un-annotated parts of genome that could lead to discovery of new genetic loci and new transcripts. The authors propose the possibility of using NGS for gene expression level quantification and comparison. They further compare expression levels obtained by Pyrosequencing with microarray analysis, reporting a correlation of 0.45. While the correlation is relatively low, it can be caused by the fact that two distinct technologies are being used (Weber et al., 2007).

Pyrosequencing was also used in one of the first studies of the Drosophila transcriptome (Torres et al., 2008). The study compares high-throughput sequencing of 3' fragments obtained by reverse transcription with restriction enzyme and random fragments sheared by nebulisation. While the former results in biases due to various fragment length, the latter produces a similar distribution of fragments for all genes. The observed technical reproducibility of expression level measurements was comparable to that of microarrays suggesting usefulness of NGS for for this type of analysis (Torres et al., 2008).

The use of RNA-seq for Single Nucleotide Variation (SNV) detection and simple differential expression analysis was reported by Sugarbaker et al. (2008). The authors study variants in expressed genes of pleural and lung cancer. Four samples of malignant pleural mesotheliomas (MPMs), two samples of pulmonary adenocarcinoma (ADCA) and a normal lung tissue were sequenced using Pyrosequencing technology. Multiple read coverage is used to detect variants in the transcribed genes leading to identification of 15 nonsynonymous variations in the MPM which are within genes that could be related to cancer. The authors further use simple log ratio test for expression of six transcripts that are known to have different abundance in MPM and ADCA. The log ratios of expression

obtained by RNA-seq correspond to previously observed ratios by real-time PCR and microarrays.

The above mentioned studies used Pyrosequencing most likely due to local availability of the technology. After 2008, the majority of studies use sequencing by synthesis by Illumina or SOLiD sequencing devices. The read length of the latter approaches is much shorter than produced by Pyrosequencing, but the two to three orders of magnitude higher number of reads eventually produces much higher coverage, improving the reproducibility of the RNA-seq expression level quantification applications.

Nagalakshmi et al. (2008) explore the RNA of yeast using sequencing by synthesis. Two fragmentation methods were used with two technical replicates and two biological replicates each, resulting in a total of almost 30 million $35bp$ reads. The authors present the applicability of RNA-seq technology for exploration of gene features and extending the known annotation, as well as for abundance quantification.

The mapped reads are used for more precise estimation of 5' and 3' ends of genes by locating the sudden changes of read coverage. Despite existing gene annotation the precise definition of the ends was lacking. Furthermore, the authors estimate gene expression by looking at the median coverage signal in a $30bp$ window located upstream of the 3' codon. 34 genes were validated using quantitative PCR with a reported correlation of 0.98. The technical and biological reproducibility of the abundance quantification using RNA-seq was also very high with 0.99 and 0.93 to 0.95 Pearson correlation coefficient respectively.

Technical reproducibility of gene expression measurements was systematically evaluated by Marioni et al. (2008), who sequenced samples from human liver and kidney tissues using Illumina's sequencing by synthesis. Each sample is represented by seven technical replicates, sequenced in two separate runs, in two different concentrations. This enables examination of the effects of using various lanes, runs and concentrations on the expression level measurements.

Gene abundances are estimated by counting uniquely mapped reads to gene's exons. Variation across lanes is reported for a small fraction of genes, resulting in a high Spearman correlation coefficient 0.96. The majority of genes have expression varying within the range of Poisson variance as it is theoretically expected. For samples sequenced at different concentrations the variance is increased for a higher number of genes.

The study further includes differential gene expression analysis with comparison to microarray technology and validation by quantitative PCR. The microarray technology was used to analyse the same two samples, reporting 8113 genes at FDR 0.1% out of which 81% were reported as differentially expressed (DE) by sequencing as well. Out of the genes reported as DE by just one of the methods, a sample of 11 genes was assayed by qPCR showing higher DE validation for genes reported by RNA-seq

The authors use a Poisson model of the count data to compare the two conditions. While the technical variance can be well estimated using a Poisson distribution for the majority of genes, the model omits natural biological variation within conditions due to the lack of biological replicates. Biological variance represents intrinsic abundance fluctuations that can cause false positive DE calls.

The report further proposes an exploratory analysis, in which previously unmapped reads are divided into two sub-reads of varying lengths and aligned to the ends of known exons. This enabled mapping junction reads, provided evidence for un-annotated splice events and demonstrated the usefulness of RNA-seq for splice variant discovery.

Another report assessing the viability of RNA-seq for gene expression measurements and splice variant detection was presented by Mortazavi et al. (2008). Here samples from mouse brain, liver and muscle tissues were assayed using sequencing by synthesis, producing 41-52 million short, 25bp, reads. The paper further introduces an application called ERANGE for estimating gene expression and detecting novel expressed sites. Mortazavi et al. (2008) introduce the RPKM expression measure, standing for Reads Per Kilobase of exon length per Million mapped reads, that was widely adopted as it attempts to represent expression in sequencing depth and gene length independent way. The relation of RPKM to read count is described in Equation 1.6.

In ERANGE the initial abundance of genes is estimated from uniquely mapping reads with multi-mapped reads being reassigned afterwards to the most probable position, also called the *rescue* method. It further detects junction reads by aligning unmapped reads to a reference of all known splice junctions and assigns them to appropriate genes. All reads assigned to a gene are then used to compute and report the RPKM value. The report shows that including the multi mapped reads improves expression level correlation with microarrays.

Apart from handling spliced reads and reads mapping to multiple locations,

ERANGE also looks for unmapped reads forming clusters outside annotated regions, which are signs of potential unannotated exons and splice sites. This way Mortazavi et al. (2008) were able to detect 596 novel candidate transcripts.

Morin et al. (2008) shift the focus on exons and apart from assessing gene abundance they survey exon-specific expression levels as well. To account for ambiguously mapping reads, the idea of *mappability* is proposed, which can be viewed as a score of uniqueness of a certain region. For each base mappability is calculated as the fraction of reads that would map uniquely out of all potential reads covering that base. The average mappability of an exon or gene is then used to divide true coverage based on unique alignments to calculate corrected coverage.

The exon coverage was validated by comparison with a custom tiling microarray. For 3908 internal exons, the Spearman correlation was 0.713, which as the authors note is comparable with correlation observed between alternative microarrays. The assessment of technical reproducibility again showed very low technical variation between sequencing runs, with 0.976 correlation of exon expression levels between replicates.

Similarly to previously mentioned studies, this report also includes exploratory analysis of novel splicing events, assessment of transcriptional start and termination sites, and single nucleotide polymorphisms. The authors also propose the idea of using RNA-seq for assaying allele-specific expression and demonstrate the possibility of fusion gene discovery by documenting reads spanning exons from independent gene loci.

All of the methods mentioned above that attempt quantification of expression levels focus on the abundance of genes. The reports acknowledge the fact that not all reads are aligned unambiguously and either resort to discarding multi-mapped reads or use some kind of correction as the rescue method used by Mortazavi et al. (2008) or mappability correction implemented by Morin et al. (2008). Despite the fact that these methods might improve the estimation accuracy, they are more of an *ad-hoc* solution.

The main drawback of these methods when it comes to expression level quantification is that they ignore the fact that reads originate from various transcripts of different lengths. As we have already discussed in Section 1.4.2, the length of a molecule directly affects the number of reads produced from a transcript and thus affects the expression of a gene. The methods above use the approach, which

counts reads from all constitutive exons of a genes. This approach can lead to spurious results in the differential expression analysis, hence combining expression levels of transcripts to estimate gene abundance should be used (Trapnell et al., 2013).

1.5.2 Methods for expression level quantification by RNA-seq

Several methods have been proposed directly addressing the problem of RNA-seq data quantification. The expression is quantified on the level of transcript isoforms of genes, which can be easily added up to calculate the total gene expression.

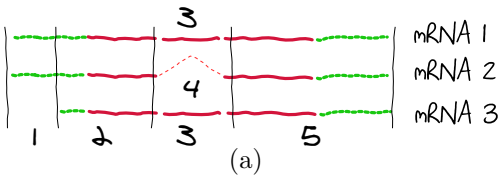
There are two ways of looking at the quantification problem, one is in terms of direct estimation, or optimisation, whereas the other perspective looks at the quantification in terms of probabilistic inference. While the former can provide an exact formulation of the problem and possibly faster algorithms, the latter approaches provide a better framework for including the inherent uncertainty of the observed data.

Non-probabilistic approaches

A similar idea to the notion of mappability proposed by Morin et al. (2008) is used in the NEUMA approach (Lee et al., 2011). NEUMA or Normalisation by Expected Uniquely Mappable Area (EUMA) combines the use of uniquely mapping reads with normalisation to estimate the expression level of transcripts. The EUMA can be thought of as effective length or a normalisation factor and is pre-calculated for every transcript by simulating all possible reads within some constrained fragment length range from all transcripts. The simulated reads are aligned to the transcriptome and the counts of reads unique to isoform, $C^{(su)}_{l_f m}$, are recorded for each fragment length l_f. EUMA is the expected number of unique reads given an experimentally observed fragment length distribution

$$EUMA_m = \sum_{l_f} P(l_f) C^{(su)}_{l_f m}. \tag{1.7}$$

The actual reads are aligned to a transcriptome reference, discarding all reads with multiple alignments and counting the unique reads per transcript, $C^{(u)}_m$. The



(a)

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

(b)

Figure 1.4: **Read counting bins and corresponding indicator matrix.** (a) Transcripts of a single gene are split into bins for counting reads. The bin boundaries are set so that every bin is either completely contained or omitted in each transcripts. Note the bin 4 will contain only reads mapping to splice junction between first and third exons. (b) The allocation of bins to transcripts is denoted in an indicator matrix, also referred to as design matrix.

expression level is expressed in terms of FVKM (Fragments per Virtual Kilobase Per Million sequenced reads), which is a similar measure to the RPKM described earlier, and is simply calculated as

$$FVKM_m = \frac{C_m^{(u)}}{EUMA_m/10^3 \times N/10^6}. \tag{1.8}$$

The EUMA and unique reads can be similarly calculated for genes and the same approach can be applied to estimate gene expression levels.

The IsoformEx method addresses the transcript expression level estimation in terms of constrained optimisation (Kim et al., 2011). The overlapping clusters of transcripts are divided into slices, which can be either exons or splice junctions, and represent regions which are shared by a unique set of transcripts. An indicator matrix A is used, where $A_{im} = 1$ denotes slice i being part of transcript m. Based on reads aligned to a genome reference, expression level in RPKM is calculated for each slice. The estimation of expression $\boldsymbol{\vartheta}$ for the overlapping cluster is then posed as a non-negative weighted optimisation

$$\arg\min_{\boldsymbol{\vartheta}} ||WA^T\boldsymbol{\vartheta} - W\boldsymbol{\vartheta}_{slice}||_2^2 \quad ; \vartheta \geq 0\,, \tag{1.9}$$

where a weight matrix is introduced to increase the importance of discriminative slices, which are exons or junctions specific to a particular transcript. The problem is identifiable and has a unique solution when the indicator matrix A is full rank. In cases when the matrix is rank-deficient, the problem can be solved for a

subset of linearly independent rows, setting the rest to 0. According to Kim et al. (2011) the observed frequency of clusters with a rank-deficient indicator matrix was below 2%, when analysing a human MCF7 cell line. The report does neither address the problem of biased read distribution nor the use of paired-end reads.

Li et al. (2011a) propose a linear model similar to the definition of the optimisation problem proposed by Kim et al. (2011). The model, implemented in the SLIDE method, is applied to individual genes for isoform discovery and quantification. The gene is again split into regions, referred to as bins, reads are counted for each bin and the read counts are normalised. The model uses a non-binary design matrix F which can be thought of as a combination of the indicator matrix A and the weight matrix W from the previous definition. Here the design matrix F_{im} defines the conditional likelihood of observing a read in the bin i given transcript m and can incorporate fragment length and positional biases. The normalised bin count of bin i is expressed as

$$\mathcal{C}_i = \sum_{m=1}^{M} F_{im}\theta_m + \epsilon_i, \tag{1.10}$$

where θ_m is the proportion of transcript m within its gene and ϵ_i is an error term. In case of isoform discovery, the model is unidentifiable and the authors use sparse estimation by Lasso regression (Tibshirani, 1996). For expression estimation, the model is identifiable and is solved by non-negative least squares optimisation.

The rQuant method proposed by Bohnert et al. (2009); Bohnert and Rätsch (2010) formulates the same problem in terms of quadratic programming optimisation. Instead of dividing genes into slices, the coverage is assessed per nucleotide. The model includes a bias correction term D_{pm} defining the read density at position p of transcript m. In case of uniform read distribution, D becomes binary indicator matrix. Basic definition of the problem is similar to those above:

$$\boldsymbol{\vartheta} = \arg\min_{\boldsymbol{\vartheta}} \sum_{p \in P} \left(C_p - \sum_{m=1}^{M} D_{pm}\vartheta_m \right). \tag{1.11}$$

The read density matrix term, D, can be further parametrised and estimated during the optimisation process.

Count-based models

Count-based models originate from an idea very similar to that of Kim et al. (2011). Reads are counted for certain regions or slices and the counts are used to estimate the expression of transcripts which contain the regions. Unlike the constrained optimisation approach used in IsoformEx, the following methods use probabilistic models based on the Poisson distribution to describe the relationship between expression levels and observed counts.

Jiang and Wong (2009) assume a uniform distribution of reads along genes and exons, in which case the number of reads coming from the exon follows a binomial distribution. They approximate the binomial distribution by a Poisson distribution and propose a statistical model for the observed data. The model does not account for reads mapped to multiple gene loci and thus only has to model the transcript proportions within each gene. For multiple transcript isoforms of a gene, the expression index of isoforms, $\boldsymbol{\xi}$, is defined as $\xi_m = \theta_m / l_m$, where θ_m would be the proportion of transcript fragments within a gene (similar to Equation 1.5) and l_m is the length of an isoform.

The exon read count C_e is modeled as a Poisson random variable representing number of reads sequenced from all transcripts containing that exon,

$$C_e \sim \text{Poisson}\left(l_e N \sum_{m=1}^{M} A_{em} \xi_m \right), \tag{1.12}$$

where $A_{em} \in \{0, 1\}$ is an indicator variable describing whether exon e belongs to transcript m. The read count for a gene is then just the sum of exon read counts. As well as proposing a maximum likelihood estimation approach which produces point estimates of $\boldsymbol{\xi}$, they also use importance sampling to sample from the posterior distribution of $\boldsymbol{\xi}$ given the data.

This model is further extended by Wu et al. (2011) into N-URD (Non-Uniform Read Distribution model) to account for non-uniform biases observed in sequencing data. Here the authors estimate a Global Bias Curve that captures the overall read distribution along genes. The Global Bias Curve estimates positional bias and is learned from reads mapping to single-isoform genes. It is applied to the model through down-weighting of exons within genes, modelling the bias on the exon scale. They also use the notion of Local Bias Curve, which is just an alternative way of denoting gene read count as a mixture of exon read counts described

by a Poisson distribution, which includes local gene-specific biases learned during the estimation.

An alternative way of using the Poisson distribution to model read counts from exons and transcripts was proposed by Richard et al. (2010). The model is implemented in the POEM method for transcript expression quantification, CASI method for detection of alternative exon usage within condition and DASI method for detection of alternative exon usage between samples from two conditions.

The POEM method is further extended in the MMSEQ application, which focuses on transcript expression level quantifications, and haplotype-specific transcript expression inference (Turro et al., 2011). The Poisson distribution is again used to model the number of reads coming from a transcript within a specific region. Most regions correspond to exons or junctions, however they are defined more loosely as partitions of transcriptome for which all reads only align to a single partition. This enables the use of reads mapping to multiple genes as these reads will just belong to a region which contains transcripts from both genes. It similarly enables defining separate regions for different haplotypes. For a region i and transcript m, the reads coming from the region are modeled as

$$X_{im} \sim \text{Poisson}\left(bl_i A_{im} \mu_m\right), \tag{1.13}$$

where b is a normalisation constant, l_i is the length of the region, A_{im} is the allocation indicator and μ_m is the expression of transcript. However the per-transcript counts from a transcripts are unobserved, the only observed variable being the total count C_i. The relation of the total count and counts coming from transcripts is given by multinomial distribution

$$(X_{i1}, \ldots, X_{iM}) \sim \text{Mult}\left(C_i, \phi_{i1}, \ldots, \phi_{iM}\right), \tag{1.14}$$

$$\phi_{im} = \frac{A_{im}\mu_m}{\sum_{m'}^{M} A_{im'}\mu_{m'}}, \tag{1.15}$$

in which reads are assigned proportionally based on the expression μ_m. The μ_m can again be regarded as a proportional measure of the number of molecules, expressed as $\mu_m = \theta_m / l_m$.

The MMSEQ and POEM models vary from that of Wu et al. (2011) in improved, probabilistic, assignment of reads within a region using the Multinomial distribution. MMSEQ further provides more flexibility in the definition of shared

regions and as the newest of all methods also accepts paired-end read data. MMSEQ enables correcting expression estimates for bias if effective lengths of transcripts are provided, where an effective length is a proportional measure of the number of reads generated from a molecule.

The MMSEQ application implements an Expectation Maximization (EM) algorithm for initial Maximum Likelihood estimation of the expression $\boldsymbol{\mu}$ and subsequently uses Bayesian inference with a Gamma prior over the expression. Gibbs sampling is used to estimated the expected value of $\boldsymbol{\mu}$ and Monte Carlo standard errors. The authors further show that the Bayesian estimate improves accuracy over the Maximum Likelihood.

All of the models based on count information use the Poisson distribution to model observed counts for a certain region. This requires summarization of the reads in terms of counts per region, which loses the information about individual reads and the qualities of their alignments. This can be prevented by use of a weighting scheme for individual reads when computing the counts, however this approach has not yet been implemented. The read bias correction imposes similar problem. Once the reads are summarized into counts, the bias can only be accounted for on the level of exons, or regions.

The obvious advantage of the count-based models is the computational speed-up. The available sequencing depth has steadily increased over last few years, resulting in experiments with billions of reads. These models avoid the necessity of dealing with individual reads in the actual inference process.

Generative models

The main distinction of the following set of methods is that they model the generative process of sequencing for each individual read. All current high-throughput sequencing technologies sequence millions of reads simultaneously, in parallel. Starting from the RNA molecules, the process of fragment preparation and sequencing is the same for each read. Overall, we can assume that generation of each read is conditionally independent of other reads, conditioned on some factors. The main factor is the initial mix of RNA molecules which determines the likelihood of a read being from a certain transcript isoform. The other factors that are shared between the reads are the properties of the actual protocol such as fragmentation biases, priming biases and fragment length filtering. The conditional independence of high-throughput sequencing of reads is utilized by the

generative models.

The RSEM model (RNA-seq by Expectation Maximization) was one of the first to use a generative model of reads (Li et al., 2010a; Li and Dewey, 2011). The input of the model are all alignments of reads mapped against the transcriptome sequence. The model is parametrised by $\boldsymbol{\theta}$, the relative abundance of fragments and defines the likelihood of the data as

$$P(\boldsymbol{g}, \boldsymbol{s}, \boldsymbol{o}, R|\boldsymbol{\theta}) = \prod_{n=1}^{N} P(g_n|\boldsymbol{\theta})P(s_n|g_n)P(o_n|g_n)P(r_n|g_n, s_n, o_n), \tag{1.16}$$

where R is the set of N reads, $\boldsymbol{g}$ is the vector indicating transcript of origin for each read and $\boldsymbol{s}, \boldsymbol{o}$ are vectors of position and strand of each read, respectively. Note that only reads are observed and the vectors describing each read are latent variables. The probability of a read being from a transcript, $P(g_n|\boldsymbol{\theta})$, is defined as a multinomial distribution parametrized by $\boldsymbol{\theta}$. The authors further define an empirical *read start position distribution* model for $P(s_n|g_n)$ and read observation likelihood $P(r_n|g_n, s_n, o_n)$ based on base mismatches. In accordance with the model, every read can originate from any arbitrary transcript and position, however for practical reasons only transcripts and positions reported by mapping algorithm are considered as otherwise the likelihood is negligible.

The RSEM method reports the maximum likelihood estimate of the expression parameter $\boldsymbol{\theta}$ which is calculated by the EM algorithm. The method was later extended to include a Gibbs Sampling algorithm, which is used for sampling from the Bayesian posterior distribution (Li and Dewey, 2011). The posterior is obtained by introducing a non-informative prior over $\boldsymbol{\theta}$ in the form of a Dirichlet distribution. The sampler is initiated by the maximum likelihood estimate and is used for inferring the posterior mean estimate and credibility intervals. The updated version also enabled the use of paired-end reads, which was not possible in the first version.

Nicolae et al. (2011) presented IsoEM a method using the same generative model principle as RSEM. Unlike RSEM, IsoEM can handle reads aligned against transcriptome as well as genomic alignments. The model is enhanced to account for paired-end data and also includes read weighting scheme similar to the one proposed by Hansen et al. (2010). The weighting scheme accounts for sequence specific biases such as that caused by random hexameter priming (Hansen et al.,

2010), but does not account for positional biases. IsoEM also reports the maximum likelihood estimate obtained by the EM algorithm, which is speeded-up by grouping reads with equivalent alignments.

Cufflinks is a tool for transcript assembly, isoform discovery, as well as isoform quantification (Trapnell et al., 2010, 2013). For the quantification of expression levels it uses a similar model to RSEM, enhanced by the ability to use the fragment length information from paired-end reads. However, in Cufflinks the model is only used to disambiguate transcripts within a genomic locus. The reads mapping to multiple genomic loci are assigned according to the *rescue* method that was described earlier and originally used by Mortazavi et al. (2008). Cufflinks estimates the maximum *a posteriori* (MAP) estimate of the expression in Fragments Per Kilobase of length per Million reads (FPKM) which is equivalent of RPKM for the case of paired-end reads. The read likelihood model of Cufflinks was improved by Roberts et al. (2011) to account for both sequence and positional biases.

The same model, with different inference algorithm, is used in the eXpress method (Roberts and Pachter, 2013). To account for the ever increasing size of sequencing data, the method uses an online EM algorithm to infer the maximum likelihood estimate of expression levels. Instead of iterating over the reads multiple times until reaching convergence, eXpress relies on the great number of reads and works on a stream of reads, using each read only once to update the expression estimate.

Nariai et al. (2013) also use a generative model, but apply a Variational Bayes (VB) inference method. The probabilistic model is based on RSEM with slight modifications. The model assumes uniform read distribution, but enables better handling of read errors, insertions and deletions. Furthermore, the read likelihood parametrisation is inferred in the initial iteration of VB inference. The approximate VB inference enables good estimation of the posterior mean, as the authors further demonstrate, and VB necessitates fewer iterations than EM algorithm before convergence.

The MISO method proposed by Katz et al. (2010) uses a generative model which assumes a uniform read distribution. The model is only applied to reads mapping to unique positions within the genome and the multi-mapping reads are discarded. Instead of quantifying the transcript expression levels, MISO estimates either the relative proportion of alternatively spliced exons or it can

estimate relative expression of isoforms within a gene locus. MISO applies a Bayesian inference approach using hybrid Markov chain Monte Carlo algorithm. The algorithm combines the Metropolis-Hastings algorithm with a Gibbs sampler to sample from the posterior distribution over the expression parameter.

The generative models provide a fine grained control over individual reads where each alignment is weighted by its likelihood, which can account for errors and biases. Furthermore, when the model is applied over all reads such as in RSEM or IsoEM, it intrinsically handles reads mapping to multiple gene loci as there is no distinction between transcripts of single or multiple genes. The natural drawback of this method is the computational complexity which scales with the number of reads in the sample.

A probabilistic generative model is also the basis of the expression estimation procedure presented in this report. The model, together with Gibbs sampling inference procedure is presented in Chapter 2. We also a present fast inference method in form of a collapsed Variational Bayes algorithm in Chapter 4.

Sequencing biases

Assuming a uniform distribution of reads along fragments simplifies the problem of expression quantification. However, biased read distributions were reported in both DNA and RNA sequencing.

Dohm et al. (2008) specifically analyse biases and errors in DNA sequencing by Illumina's sequencing by synthesis. The report shows biases towards regions with higher GC content, higher distribution of base errors close to read ends and base errors preceded by specific sequences.

In the case of RNA-seq, the extraction and preparation of RNA have further effects on the final read distribution, which have been reported in some of the earliest studies using RNA-seq. Mortazavi et al. (2008) observed biased representation and under-representation of certain sites in random priming of cDNA, which were reduced by fragmentation of the RNA. In other protocols, higher representation of 3' ends over 5' ends was observed, most likely due to enrichment of 3' sequence in the purification step and poly-T priming (Nagalakshmi et al., 2008). Conversely, over-representation of 5' ends was reported by Morin et al. (2008), who also reported under-representation of inner exons in respect to terminal exons, which can be avoided by sonicating the random primed cDNA. While some biases can be avoided with improvements of the sequencing protocol,

the read distribution in most experiments is still not uniform. The importance of non-uniform read distribution for accurate expression quantification led to reports specifically focused on the exploring the biases and methods for their correction.

Hansen et al. (2010) investigated sequence specific biases occurring in Illumina's sequencing by synthesis. The report analyses multiple independent datasets that were sequenced on Illumina's Genome Analyser and confirms consistent patterns in nucleotide frequencies. The patterns are observed at the beginning of reads, with up to the first 13 nucleotides being affected. These effects are caused by random hexameter priming used for reverse transcription. Comparison with DNA and ChIP-seq sequencing datasets, that do not contain these biases, further confirms a cause specific to the RNA-seq protocol.

The authors further consider a read weighting scheme that can be used to adjust observed read counts in order to generate more uniform transcript coverage. The weights are calculated using estimated proportions of heptamers for each position along a read sequence. For a read starting with heptamer h, the weight is a fraction of frequency of the heptamer at the last 6 positions over its frequency at first two positions. Instead of counting reads over a region, the sum of their weights is calculated and used as the read count.

The advantage of this method is its applicability to quantification models that are built on top of counts of read per region. However, one has to be careful when applying this method in cases when the model relies on the Poisson properties of the count data. While this method addresses the problem of what we refer to as *sequence specific* bias, there are also *position specific* biases which result in increased coverage of certain regions of transcripts.

The sequence specific biases are analysed also in the report by Li et al. (2010b). Here the authors use multiple adaptive regression trees to learn the sequence dependent sequencing preferences based on highly expressed genes. The model is subsequently used to adjust transcript expression levels by dividing it by the sum of sequencing preferences of each transcript. The sum of sequencing preferences can be thought of as the effective length of a transcript or a proportional measure of the output of reads from a transcript.

Position specific bias was considered by Howard and Heber (2010). The report analyses data sequenced using sequencing by synthesis with RNA fragmentation by sonication and data sequenced using pyrosequencing, where the cDNA was fragmented by nebulisation. Both datasets show biased read distribution along

transcripts. While the first data show under-representation of both ends of transcripts, the latter data show over-representation of 5' end and a spike in coverage before the 3' end. These results show protocol dependent effects that have to be estimated on a per-experiment basis.

The authors presented a count based method for transcript quantification similar to that of Jiang and Wong (2009) improved by accounting for non-uniform read coverage of transcripts. They use kernel density estimation (KDE) to model the read coverage based on empirical distribution of reads mapping to well annotated transcripts. The coverage density is then used to weight mapping likelihoods of reads assigned to shared regions.

The N-URD expression estimation model by Wu et al. (2011) mentioned above similarly accounts for overall positional biases. The Global Bias Curve empirically estimates coverage along transcripts. It is then used to adjust the per-exon read counts that are used for expression inference.

As an improvement to the Cufflinks suite for transcript discovery and expression quantification, Roberts et al. (2011) proposed a bias correction method that aims to account for both sequence specific and positional biases. Here the likelihood of each alignment is adjusted by a bias weight normalised by the sum of bias weights over the length of a fragment. The bias weight itself contains two factors: the positional preference weight and the sequence specific weight.

The bias model is empirically estimated using only the uniquely mapping reads. While the positional and sequence specific biases can confound each other, the model is simplified and the two biases are learned independently.

For sequence specific bias a variable length Markov model is used to estimate the likelihood of observing a read from a position depending on 21 surrounding nucleotides. This is calculated as the ratio of the observed frequency of specific nucleotides over the frequency of the nucleotides under a uniform model where each position is sequenced with the same probability.

The positional bias weight is computed for fixed number of bins along transcripts, again as a ratio of observed frequency of reads from a specific bin over the frequency under a uniform model.

The bias weights are parameters of the model that have to be inferred as they are expression dependent. However, Roberts et al. (2011) report that the bias weights change marginally after initial expression estimation. Hence it is possible to resort to an approximation where the bias weights are estimated with respect

to expression levels inferred assuming uniform read distribution and subsequently used to infer corrected expression levels.

Further details of this bias correction approach are presented in Section 2.2.5.

1.5.3 Differential expression analysis methods

One of the main reason to quantify expression levels is to enable a comparison of various conditions in terms of significant differences in gene expression. The comparison of expression levels between different conditions can reveal gene functions and the mechanisms controlling their transcription. The DE analysis has been studied from various perspectives. One can either consider a gene locus as a unit of interest, focus on exon usage or compare the abundances of particular transcripts.

Gene level analysis

The most common comparison of abundances is performed on the gene level. The abundance of a gene can be summarized by the number of reads aligning to the gene and these are compared between conditions.

The Poisson distribution has been widely used for modelling the expression within a single experiment. However, due to its variance always being equal to the mean, it cannot account for over-dispersion caused by biological variations of abundances within one condition. The negative binomial distribution provides a natural extension to the Poisson distribution with a better control over the variance and was originally applied for DE analysis of SAGE data (Robinson and Smyth, 2008).

Models based on the negative binomial distribution were proposed in multiple methods for RNA-seq DE analysis. edgeR (Robinson et al., 2010), DESeq (Anders and Huber, 2010) and baySeq (Hardcastle and Kelly, 2010) all use negative binomial distribution to model the read count of a gene g within a single replicate r,

$$C_g^{(r)} \sim \mathrm{NB}(N^{(r)}\theta_g^{(c)}, \phi_g^{(r)}), \tag{1.17}$$

which is parametrized by its mean $N^{(r)}\theta_g^{(c)}$ and the dispersion parameter $\phi_g^{(r)}$, where the variance of the distribution is $Var(C_g^{(r)}) = N^{(r)}\theta_g^{(c)}(1 + N^{(r)}\theta_g^{(c)}\phi_g^{(c)})$. Here c denotes a particular condition, $N^{(r)}$ is the total number of reads for experiment r, $\boldsymbol{\theta}^{(c)}$ is the relative abundance of gene within condition c. These methods

differ in the ways they handle the dispersion and call differential expression.

The edgeR method uses conditional maximum likelihood to estimate a gene-specific dispersion parameter ϕ_g, which is shared through all the replicates. The differential expression is then determined using an exact test previously described by Robinson and Smyth (2008).

In DESeq the authors propose the use of smooth function $\nu^{(c)}$ that is shared by all genes and defines the dispersion in terms of relative expression as $\phi_q^{(c)} = \frac{\nu^{(c)}\left(\theta_g^{(c)}\right)}{(\theta_g^{(c)})^2}$. Here the differential expression is again assessed using exact test similar to edgeR and the one proposed by Robinson and Smyth (2008).

In baySeq, the parameters of the negative binomial distributions are shared within equivalence classes. If a gene is not differentially expressed, the observed counts for that gene come from one equivalence class, hence having the same parameters. For differentially expressed genes the replicates from conditions that differ belong to different equivalence classes and have independent parameters $\theta_g^{(c)}$ and $\phi_g^{(c)}$. An Empirical Bayes procedure is used to calculate the posterior probabilities of models defining the equivalence classes.

Apart from the negative binomial model, alternative methods have been proposed as well. The DEGseq method (Wang et al., 2010b) uses Binomial distribution to model the counts and tests for differential expression based on the observation that the log fold change, conditioned on the log mean expression, approximately follows Normal distribution. However, the method does not address the problem of biological variation mentioned earlier.

Auer and Doerge (2011) proposed a two staged Poisson model (TSPM) approach, which uses Poisson distribution with quasi-likelihood approach to account for over-dispersion. The method firstly filters out low expressed genes in the first stage and then assesses the differential expression for the genes that have sufficient expression.

The performance of TSPM method as well as edgeR, DESeq and baySeq is compared in a report by Kvam et al. (2012). More general review of tools for RNA-seq analysis and an overview of some of the early DE analysis methods can be found in (Oshlack et al., 2010).

As we have mentioned above, these methods use the counts of reads mapping to a particular gene to detect differentially expressed genes. They assume that the data can be approximated by the Poisson model with additional dispersion due to

biological variation. However, genes undergo splicing which can result in various transcripts with varying lengths. The fact that genes can have different lengths in different conditions, or that the relative proportions of gene's transcripts can vary cannot be accounted for in these models.

Analysis of splicing events

As RNA-seq experiments sample entire transcripts it is possible to analyse exon usage and the variation in splicing events. Unlike the methods mentioned above, which only compare the abundance of genes within a sample, one can study the variation in splicing events of individual genes.

Singh et al. (2011) proposed a method based on *Flow difference metric* (FDM) which compares the splicing variations of genes between conditions. Instead of relying on annotation and quantifying individual transcripts, the method uses a weighted splice graph representation of mapped reads. A component of the graph corresponds to a non-overlapping transcribed region with transcription start and end sites being the nodes of the graph. Edges correspond to either exons or junctions and are weighted by the amount of reads mapping to them.

The authors proposed the FDM to measure the splicing activity of a gene within condition and use a non-parametric test for the hypothesis that the FDM of two conditions are significantly different. This approach does not allow careful examination of splice variants present in each sample, but it can determine whether the splicing of a gene is different or not.

Splicing of genes can be also examined in terms of the cassette exons. These are exons that have switch like behaviour of either being included within a transcript sequence or being excluded during splicing. The Multivariate Analysis of Transcript Splicing (MATS) method proposed by Shen et al. (2012) is aimed at detecting changes in the cassette exon usage between conditions.

Here the authors use read counts of junctions to calculate the exon inclusion level. The reads mapping to the junction of particular exons with other upstream or downstream exons are counted as the exon's inclusion reads, I. On the other hand, the reads mapping to junction of surrounding exons are counted as exon skipping reads, S. The exon inclusion level is defined as the ratio of inclusion reads over all related reads, $\hat{\phi} = I/(I + S)$. The exon inclusion level is used to construct a prior distribution for overall splicing similarity, which can be subsequently combined with a Binomial model of read counts to infer the posterior

distribution of alternative splicing of genes.

Similarly to the FDM method, MATS enables the detection of alternative splicing between conditions. Additionally it enables a comparison of individual exon usage. The downside of this method is that it only relies on the reads mapping to junctions of exons. While the method does not directly use a bias correction approach, the report suggests using weighted counts based on of the previously mentioned techniques.

The DEXSeq method similarly focuses on the differences in exon usage between various conditions (Anders et al., 2012). Gene loci are split into bins with unique alignments and the analysis is carried out using the per-bin read counts. The reads spanning junctions of bins are assigned to both bins and junctions overlapping exons are not considered as evidence of exclusion. The approach is very similar to that of DESeq tool for gene DE analysis, previously proposed by Anders and Huber (2010), co-authors of DEXSeq.

The bin read count is modeled by a negative binomial distribution with mean equal to the product of the library size and the relative proportion of fragments from the bin. The bin's relative proportion of fragments is a product of the following factors: gene expression, expected fraction of reads within the gene mapping to the bin, fold change of the gene between conditions and condition specific effect on bin's abundance. The last two factors are condition specific and can be used to assess the DE of a particular gene and to estimate the difference in bin usage between conditions.

Most of the bins equal to single exons, however the authors do not address the case when one exon contains multiple bins. Another downside of this method is that it considers exons independently, and as authors note themselves, in cases when multiple exons exhibit DE it can be indistinguishable whether the change is caused by alternative splicing or difference in abundance.

The above mentioned approaches for analysis of alternative splicing can be used to detect when a gene is being spliced differently between conditions. While MATS and DEXSeq also quantify the exon usage, none of these methods is designed for comparing changes in the abundance of particular isoforms or transcripts. Nevertheless, their advantage lies in the fact they do not rely on correct annotation of transcripts and only need splice-aware alignment of reads.

Transcript level analysis

Differential expression analysis of transcripts provides the most detailed assessment of the differences between conditions. While gene level analysis can be confounded by varying splicing, exon level analysis only provides evidence of differences in the splicing process. The following approaches use RNA-seq data to determine the significant changes in abundances of particular gene transcripts.

These approaches fall into two categories, the first type of methods uses aligned reads as an input, and the expression levels of transcripts in individual samples can be only latent variables of the model. The second type relies on an additional transcript quantification tool and uses transcript expression levels from individual experiments as an input for the analysis.

Zheng and Chen (2009) proposed a method for DE analysis using either RNA-seq read coverages or probe intensities obtained by tiling microarrays. The Bayesian Analysis of Splicing Isoforms (BASIS) method is based on hierarchical probabilistic model of observed changes in expression. The expression of a 'probe', which can be either microarray probe or read coverage of a genomic position is defined as a mixture of expressions of transcripts that 'cover' the probe. Instead of defining a model of the observed coverage, the authors use the multinomial Normal distribution to model the difference of probe coverage between conditions. A Markov chain Monte Carlo algorithm is used to sample from the posterior distribution of model parameters.

This method uses the Normal distribution to model the difference of expression. While this might be a good choice for intensities of probes, it is not suitable for modelling differences in discrete read counts. Furthermore, the method does not address read distribution biases that can skew the observed read coverage. However, the most important disadvantage of the approach is that it does not allow to include biological replicates that would enable assessment of the biological variability.

Drewe et al. (2013) proposed two complementary methods for transcript differential expression analysis in an application called rDiff. In the case of a known annotation, a parametric model based on a negative binomial distribution of observed counts is used. For cases when the exact transcript annotation is unknown a non-parametric approach using kernel maximum mean discrepancy can be applied.

The parametric approach relies on 'alternative regions' which are regions within genes that are unique to single transcript. The statistical test considers each region independently to determine whether a transcript containing that region is differentially expressed. The null hypothesis is that there is no differential expression and that counts observed in replicates of each condition are realized from a negative binomial distribution with the same parameters. Based on the negative binomial model a p-value of the observed counts under the null hypothesis can be calculated for each region. The Bonferroni correction method is used to combine p-values from multiple regions into a single p-value of a transcript.

The rDiff method uses biological replicates to estimate the variance of normalised counts. Observed empirical variance of normalised counts is processed by local regression producing a mapping between mean normalised read count and biological variance.

The Cufflinks application suite for transcript assembly and quantification also includes the Cuffdiff method for differential expression assessment (Trapnell et al., 2010, 2013). The transcript expression inferred by Cufflinks algorithm is used in a negative binomial model that accounts for over-dispersion, similarly to the DESeq method by Anders and Huber (2010). Here authors acknowledge the ambiguity of read assignments in transcript quantification that makes it inappropriate to use negative binomial distribution over transcript read counts. They propose the use of beta negative binomial mixture based on subsets of transcripts that share mapped reads. This enables estimation of expression levels as well as variance and covariance of expression between transcripts.

The parameters of the beta negative binomial mixture for each condition are calculated exactly based on moments of the distribution. Two-sided test statistic of log ratio of gene expression divided by variance is used to assess gene level differential expression. To assess the significance of changes in relative proportions of transcripts authors use square root of Jensen-Shanon divergence of the relative abundances in two conditions as a test statistic. The p-value is calculated empirically by sampling from the distributions of relative abundances under the null hypothesis of no DE.

EBSeq method uses estimated transcript read counts combined with negative binomial model to detect differentially expressed transcripts and genes (Leng et al., 2013). The report suggests using estimated read counts from RSEM, Cufflinks or any other quantification method as an input. The read count of

a transcript m within a single replicate r is modeled by the negative binomial with mean $\mu_m^{(r)} = n^{(r)}\rho_m(1 - q_m^{(c)})/q_m^{(c)}$ and variance $(\sigma_m^{(r)})^2 = \mu_m(r)/(q_m^{(c)})$, where $n^{(r)}$ is a replicate specific normalisation constant and ρ_m is a transcript specific parameter shared over all conditions. The model assumes a beta distribution prior over the condition specific parameter $q_m^{(c)} \sim \text{Beta}(\alpha, \beta_G)$, with constant hyperparameter α and varying hyperparameter β_G, dependent on the number of splice variants belonging to the same gene.

The method uses empirical Bayes inference procedure where the maximum likelihood estimates of hyperparameters are inferred from the model using the EM algorithm. The hyperparameter estimates are subsequently used in the inference of the posterior distribution of differential expression, i.e. the probability of $q_m^{(c1)} \neq q_m^{(c2)}$.

1.5.4 Other applications of RNA-seq

RNA-seq can be used in various types of studies, with the most obvious application being replacement of the microarray technology in gene expression related research. For some types of experiments, such as splice variation discovery, the sequencing based approach is much more convenient. With the sufficient depth of sequencing, multiple-fold coverage of the underlying sequence can be obtained. This enables transcriptome analysis of species that have not been annotated or might even have unknown genome.

De-novo assembly

Similarly to whole genome sequencing, RNA-seq can be applied to organisms with an unknown genome. The transcripts have to be assembled de-novo using just the short reads. While the task is very similar there are differences that have to be considered. Firstly, the transcriptome does not contain long repetitive regions as are found in genome. The transcriptome is also an order of magnitude shorter than genomic sequence. On the other hand, isoforms of the same gene can be very similar making it hard to distinguish between them. Lastly, unlike genome sequencing, the coverage of transcripts is uneven, depending on their abundance. While transcripts of one gene might be highly covered, low expressed transcripts are hard to identify.

Tools specifically designed for de-novo discovery of transcripts are available,

e.g. Trans-ABySS (Robertson et al., 2010) and Velvet (Zerbino and Birney, 2008).

Transcript discovery using the reference genome

A much simpler and more common task is the identification of new transcripts by RNA-seq for organisms with a known genome. Instead of assembling the reads to continuous sequence, the genomic sequence is used as a reference and most of the reads can be aligned to the genome. However, reads providing most information about gene isoforms are reads that span splice junctions.

Reads from exon junctions, or 'junction reads', cannot be aligned to the genomic sequence because the reference contains also introns which were spliced out. These reads have to be aligned by first aligning only smaller parts of the reads. Once a part of a read is matched to a position within reference, the alignment can be extended as long as the read matches reference, the rest of the read has to be re-aligned as it is part of a different exon. The spliced alignments provide two-fold information. Firstly they mark exactly exon boundaries and secondly they tell us which exons are being spliced together.

Once the reads are aligned, consecutive sequences covered by reads are identified as exons. Some genes can also be identified based on groups of exons that are positioned close to each other. Finally, using the identified exons and the information gained from the spliced reads, a set of all potential transcripts can be constructed. Exact identification of true transcripts is in most cases impossible. While some applications report all possible transcripts (Guttman et al., 2010) and leave the user to assess their validity, other applications try to identify the smallest set of transcripts explaining all the reads (Trapnell et al., 2010).

In transcript discovery the use of paired-end reads can provide additional information that can help deciphering various gene isoforms. The mates of a paired-end read were sequenced from one fragment which was size selected. This means that the reads are from the same transcript and have to be aligned within a certain distance of each other. Paired-end reads from fragments spanning a junction thus provide the same kind of information as junction reads except for exact exon boundaries.

Transcript discovery is a well studied problem with many existing applications such as the popular Cufflinks suite (Trapnell et al., 2010), Scripture (Guttman et al., 2010), Trinity (Grabherr et al., 2011) or IsoLasso (Li et al., 2011b).

Even for annotated organisms, new isoforms of genes are still being discovered. The annotation provides the advantage of a well defined exons and their boundaries, which simplifies the alignment. However, for a set of n exons, there are $2^n - 1$ potential splice variants and the problem of selecting the correct subset of isoforms still preserves. Some tools such as Cufflinks can use previously annotated transcripts and propose new transcripts only for reads that do not align with existing annotation (Trapnell et al., 2010). On the other hand, SLIDE (Li et al., 2011a) depends solely on reads and annotation.

For a review of methods for transcript discovery please refer to Martin and Wang (2011) and Garber et al. (2011).

Fusion gene detection

Similarly to discovering splice isoforms of genes, RNA-seq can be used for detecting fusion genes. These are transcripts that were created by fusion of exons of different genes. These transcripts occur in various types of cancer and their discovery is important for further detection and understanding of various types of cancer (Maher et al., 2009).

Allele specific expression

Most higher organisms have multiple copies, or alleles, of each gene, due to having two or more of each chromosome. RNA-seq can be used to identify allele specific transcripts and quantify them (Turro et al., 2011; Skelly et al., 2011). Condition specific prevalence of a certain allele is an interesting problem which can be transformed to the problem of transcriptome quantification. Instead of deciphering transcripts which differ in splice variation, one has to distinguish between transcripts with similar sequence that differ just in several SNPs.

1.5.5 Bayesian inference

Probabilistic modeling

Bayesian inference encompasses methods of statistical inference that rely on the application of Bayes' theorem to update beliefs based on observed evidence. Bayes' theorem

$$P(\psi|\mathcal{D}ata) = \frac{P(\mathcal{D}ata|\psi)P(\psi)}{P(\mathcal{D}ata)}\,, \tag{1.18}$$

relates our belief of some unknown parameter ψ after observing evidence $\mathcal{D}ata$ to our prior belief of ψ, the likelihood of the $\mathcal{D}ata$ given ψ and the overall likelihood of the observed $\mathcal{D}ata$. We use the following terms when describing the model:

- $P(\psi|\mathcal{D}ata)$ is the *posterior* distribution, or our belief of ψ after observing the $\mathcal{D}ata$;
- $P(\mathcal{D}ata|\psi)$ is the *likelihood* of the observed $\mathcal{D}ata$ given the parameter ψ;
- $P(\psi)$ is the *prior* distribution expressing our initial belief of ψ;
- $P(\mathcal{D}ata, \psi) = P(\mathcal{D}ata|\psi)P(\psi)$ is the *joint likelihood* of the observed $\mathcal{D}ata$ and parameters ψ;
- $P(\mathcal{D}ata)$ is the *marginal likelihood* of the $\mathcal{D}ata$, expressing the overall probability of observing the $\mathcal{D}ata$ based on our model irrespective of the parameter ψ.

The marginal likelihood can be also viewed as a marginalization of the joint likelihood $P(\mathcal{D}ata) = \int \mathrm{d}\psi \, P(\mathcal{D}ata, \psi)$.

The key concept of Bayesian inference is that all parameters are represented by probability distributions. This is unlike alternative *frequentist* approaches in which a single value of a parameter is estimated, with an additional measure of certainty. The use of probability distributions and Bayes' theorem provide a natural framework for combining the certainty of our beliefs and propagating them throughout the inference. The inferred distributions can be used in further analysis or simply summarized by their moments.

Model inference

The inference procedure starts by defining a probabilistic model of the observed $\mathcal{D}ata$, which defines the likelihood of the data in terms of the unknown parameter, $P(\mathcal{D}ata|\psi)$. We then select a prior distribution over the model parameters, $P(\psi)$, which can either represent some information we have about the parameters or provide an *uninformative* base distribution for the parameters. After defining the model and selecting a prior, we can apply Bayes' theorem to derive the posterior distribution over ψ,

$$P(\psi|\mathcal{D}ata) = \frac{P(\mathcal{D}ata|\psi)P(\psi)}{\int \mathrm{d}\psi \, P(\mathcal{D}ata|\psi)P(\psi)} \,. \tag{1.19}$$

While this approach can be universally applied to any kind of model and prior, the marginal likelihood in the denominator is in most cases intractable. The marginal likelihood serves as a normalisation constant as it is not a function of the parameter ψ. In cases when the marginal likelihood is not tractable, we can derive a proportional version of the posterior distribution, without knowing the normalisation,

$$P(\psi|\mathcal{D}ata) \propto P(\mathcal{D}ata|\psi)P(\psi)\,, \tag{1.20}$$

and use an approximate inference method to infer the parameters. There are two main classes of approximate inference, asymptotic approximation methods represented by the Markov chain Monte Carlo algorithms and deterministic approximations, which approximate the posterior with tractable probability distributions.

Markov chain Monte Carlo

The Markov chain Monte Carlo (MCMC) algorithm refers to a set of methods which enable generating samples from a desired distribution using a Markov chain. The Markov chain is a random process, or a sequence of random variables $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$, in which the next state $\boldsymbol{x}^{n+1}$ only depends on the current state $\boldsymbol{x}^n$:

$$P(\boldsymbol{x}^{n+1}|\boldsymbol{x}^n, \boldsymbol{x}^{n-1}, \ldots, \boldsymbol{x}^1) = P(\boldsymbol{x}^{n+1}|\boldsymbol{x}^n). \tag{1.21}$$

If a Markov chain is irreducible and aperiodic, it converges to a marginal distribution $P(\boldsymbol{x}^n)$. Through careful selection of the transition probability $P(\boldsymbol{x}^{n+1}|\boldsymbol{x}^n)$, we can ensure that the marginal distribution $P(\boldsymbol{x}^n)$ is some desired distribution of our choice.

Metropolis-Hastings

In the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) a sample or state of the Markov chain at step t, $\boldsymbol{x}^t$, is first generated from a proposal distribution $Q(\boldsymbol{x}^*|\boldsymbol{x}^{t-1})$ conditioned on the last sample $\boldsymbol{x}^{t-1}$. The newly proposed sample is then accepted as a new sample $\boldsymbol{x}^t$ with acceptance probability p_{accept}. In case of a rejection, the last sample $\boldsymbol{x}^{t-1}$ is used again as a sample from the Markov chain. Hastings (1970) showed that setting the acceptance probability to

$$p_{accept} = \min\left(1, \frac{P(\boldsymbol{x}^*)Q(\boldsymbol{x}^{t-1}|\boldsymbol{x}^*)}{P(\boldsymbol{x}^{t-1})Q(\boldsymbol{x}^*|\boldsymbol{x}^{t-1})}\right) \tag{1.22}$$

ensures that the samples of the Markov chain will converge to the desired distribution $P(\boldsymbol{x})$. Importantly, p_{accept} contains the ratio of the two posteriors, which means that the normalisation constant cancels out, thus we can sample from $P(\boldsymbol{x})$ using its proportional expression as in Equation 1.20.

Selecting the right proposal distribution for Metropolis-Hastings algorithm is very important. The ideal case is to use the posterior distribution, which leads to all samples being accepted, because $p_{accept} = 1$. However, we resort to this algorithm when we are not able to sample from the true posterior, thus need a different way of proposing new samples. Metropolis et al. (1953) originally proposed the use of a symmetric proposal distribution, such as the Normal distribution centered around last sample $\boldsymbol{x}^{t-1}$. Such a procedure is also referred to as the Random walk MCMC. The symmetric distribution simplifies the acceptance probability into the ratio of posteriors:

$$p_{accept} = \min\left(1, \frac{P(\boldsymbol{x}^*)}{P(\boldsymbol{x}^{t-1})}\right). \tag{1.23}$$

Gibbs sampling

The Gibbs sampling algorithm (Geman and Geman, 1984), is a special case of a Metropolis-Hastings algorithm. Here conditional distributions of individual parameters are used to propose new values of the parameters:

$$Q(\boldsymbol{x}^*|\boldsymbol{x}^{t-1}) = \prod_{k=1}^{K} P(x_k^*|x_1^*, \ldots, x_{k-1}^*, x_{k+1}^{t-1}, \ldots, x_K^{t-1}). \tag{1.24}$$

Given the above proposal, the acceptance probability of a sample is always 1, hence all samples are accepted, providing a great advantage to the Gibbs sampling algorithm. Nevertheless, in order to use the Gibbs sampling, one has to derive conditional distributions over the parameters in a tractable form of some standard probability distribution.

Convergence of MCMC

MCMC represents an asymptotic approximative algorithm, because it converges to the desired distribution asymptotically. As we can only run MCMC for a finite number of iterations, convergence of the algorithm has to be monitored in order to ensure sufficient convergence to the marginal distribution. Furthermore, due to the Markov property of the generated samples, the sequence of samples

usually exhibits a certain level of autocorrelation. This means that despite convergence and sampling from the desired distribution, a small number of samples might not properly represent the full distribution. Various techniques for monitoring convergence and assessing the quality of generated samples can be found in related literature (Gilks et al., 1995; Gelman et al., 2004).

Deterministic approximative inference

While the MCMC algorithm asymptotically samples from the true posterior, ensuring convergence of the algorithm can be prohibitively computationally expensive. In such cases, deterministic approximate inference methods can be used instead. These methods use standard parametric probability distributions to approximate the intractable posterior distribution of the model.

We apply the approach of Variational optimisation, also known as the Variational Bayes (Bishop, 2006), to the expression quantification problem in Chapter 4. For details about alternative approaches such as the Laplace approximation or Expectation Propagation, please refer to appropriate literature (Bishop, 2006).

Chapter 2

Inferring transcript expression

In this chapter we address the problem of transcript expression quantification from RNA-seq data defined in Section 1.4.2. In brief, given a set of reads that were sequenced from an unknown mixture of transcript molecules we want to infer the abundance of the molecules within the sample.

We propose a probabilistic generative model of the RNA-seq data similar to that of Li et al. (2010a). The model is based on the known sequencing process and assumes conditional independence of reads, conditioned on a fixed set of transcripts and their unknown abundance. The model accounts for paired-end reads, fragment length distribution, base errors and read distribution biases by estimating the likelihood of each alignment.

We use a Bayesian inference approach and derive a posterior distribution over the expression parameter $\boldsymbol{\theta}$. The exact posterior is not analytically tractable, hence we apply the Gibbs sampling algorithm to generate samples from the distribution. The inferred distribution can either be used in the downstream analysis or can be summarized by calculating the mean expression.

The model is evaluated using both synthetic and real RNA-seq data. We examine the properties of the posterior distribution and assess the accuracy of the inferred expression. Finally, a comparison with other state-of-the-art methods used for transcript expression quantification is presented.

2.1 Probabilistic model of RNA-seq

2.1.1 Relative proportion of transcript fragments

The model of the observed RNA-seq data is defined with respect to the main unknown parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$, the relative proportion of transcript fragments. The use of the relative proportion of transcript fragments is more convenient for us than defining the model in terms of relative abundance of entire molecules. The relation between the abundance of transcript molecules and the number of its fragments is expressed by the effective length of a transcript, $l_m^{(eff)}$. The effective length, which depends on the length of a transcript, fragment length distribution and sequencing biases, is defined later in Section 2.2.6.

$\boldsymbol{\theta}$ can be also viewed as the normalised per-transcript read count and can be easily transformed into alternative measures. Given the effective length $l_m^{(eff)}$ and the total number of reads N, we can transform the inferred $\boldsymbol{\theta}$ into read count or RPKM as follows:

$$C_m = \theta_m \times N, \tag{2.1}$$

$$RPKM_m = \frac{\theta_m}{l_m^{(eff)}} \times 10^9. \tag{2.2}$$

2.1.2 Read-centric view of the sequencing process

In order to introduce the generative model we take an alternative view of the sequencing procedure from the perspective of a single read. Each read is generated by an independent random process and the high-throughput sequencing technology can perform millions of these processes simultaneously.

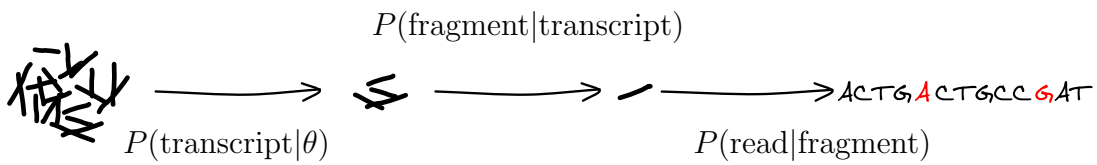


Figure 2.1: **Diagram of sequencing as an independent process generating a single read.**

The random process of sequencing a single read is schematically outlined in Figure 2.1. Each read is sequenced from one of the fragments within the sample, while not all of the fragments are necessarily sequenced. At this point we do not consider how the transcripts were fragmented, we only consider the number

of fragments from each transcript and the fact that the relative proportion of fragments is $\boldsymbol{\theta}$, the unknown parameter that represents expression.

Starting from a sample consisting of fragments of all the molecules, one of the fragments is selected with a probability $P(\text{fragment}|\theta)$. As we outline in Figure 2.1, this probability can be further divided into two factors

$$P(\text{fragment}|\boldsymbol{\theta}) = P(\text{fragment}|\text{transcript})P(\text{transcript}|\boldsymbol{\theta}).$$

In the first step, all fragments from a particular transcript are chosen with a probability proportional to $\boldsymbol{\theta}$, which is given by a Categorical distribution. Subsequently a particular fragment is chosen from all the fragments of this transcript with a probability $P(\text{fragment}|\text{transcript})$. If we assume that transcripts fragment uniformly along their length, then the probability of choosing a fragment starting at a specific position is equal to $1/(l_m - l_{\text{fr}} + 1)$, where l_m is the length of the transcript and l_{fr} is the length of the fragment. Reads observed in most RNA-seq experiments do not have a uniform distribution, meaning that transcripts tend to be fragmented at some positions more often than at others. We discuss the inclusion of a read-distribution bias model in the probability $P(\text{transcript}|\text{fragment})$ in section 2.2.

Once the fragment has been chosen, the actual read is sequenced from one of its ends with a probability $P(\text{read}|\text{fragment})$ that models the probability of base mismatches and other errors observed within the data. In case of paired-end reads, both ends of the fragment are sequenced resulting in two *mates*.

The probability of observing a single read given $\boldsymbol{\theta}$ can be written as:

$$P(\text{read}|\boldsymbol{\theta}) = P(\text{transcript}|\boldsymbol{\theta})P(\text{fragment}|\text{transcript})P(\text{read}|\text{fragment}), \quad (2.3)$$

with the important property of factorization into the probability of choosing a transcript and the probability of generating a read from the transcript. We use this property in the formal generative model of the data described in the following section.

2.1.3 Generative probabilistic model

We describe the model in Figure 2.2(a) using the standard plate notation. Sequencing generates N reads, which are the data we observe, denoted by random variables

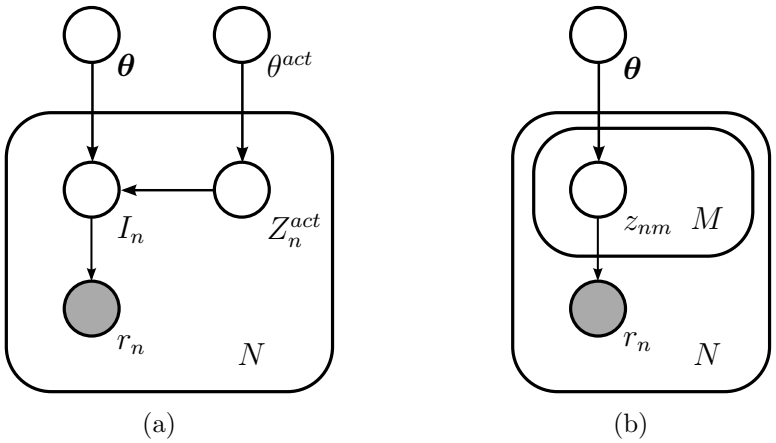


Figure 2.2: **Graphical model of the sequencing process.** Observed reads (r_n) are conditionally independent of expression ($\boldsymbol{\theta}$), given their assignment to transcripts (I_n or $\boldsymbol{z}_n$). (a) representation with explicit noise model and indicator variable I_n used for read assignments (b) noise transcript is included with the other transcripts, binary allocation vector $\boldsymbol{z}_n$ is used for read assignments.

$R = \{r_1, \ldots, r_N\}$. Each read is sequenced from a particular transcript and we use an indicator variable $\boldsymbol{I} = (I_1, \ldots, I_N); I_n \in \{0, \ldots, M\}$ to specify this assignment. The main parameter of interest is $\boldsymbol{\theta}$, which denotes the relative proportions of transcript fragments in the sample. The reads are conditionally independent of $\boldsymbol{\theta}$ given the latent variable $\boldsymbol{I} = (I_1, \ldots, I_N)$.

The model further defines a *noise* parameter θ^{act}. As some of the reads might be of low quality or have ambiguous alignments with low probability, the noise parameter governs a probability of assigning these reads to a *noise* transcript. Another latent variable $\boldsymbol{Z^{act}} = (Z_1^{act}, \ldots, Z_N^{act})$ is introduced to indicate whether a read is assigned to noise $Z_n^{act} = 0 \Rightarrow I_n = 0$ or whether it is assigned to one of the transcripts $Z_n^{act} = 1 \Rightarrow I_n \in \{1, \ldots, M\}$.

The joint likelihood of the model is

$$P(R, \boldsymbol{I}, \boldsymbol{Z^{act}}, \boldsymbol{\theta}, \theta^{act}) = P(\boldsymbol{\theta})P(\theta^{act}) \prod_{n=1}^{N} \left(P(r_n|I_n)P(I_n|\boldsymbol{\theta}, Z_n^{act})P(Z_n^{act}|\theta^{act}) \right). \tag{2.4}$$

The noise indicator variable can have values 1 and 0 and determines whether a read is assigned to one of the transcripts or to the noise transcript, and follows a Bernoulli distribution with the parameter θ^{act}. Given that $Z_n^{act} = 1$ the read is assigned to one of the transcripts, which is equivalent to 'choosing a transcript',

or $P(transcript|\boldsymbol{\theta})$ as described in the Section 2.1.2. The probability of choosing a specific transcript is proportional to the abundance of fragments of that transcript, leading to the indicator variable I_n being distributed according to a Categorical distribution with a parameter vector $\boldsymbol{\theta}$. The likelihood of observing a particular read given its assignment to a transcript, $P(r_n|I_n)$, is discussed in detail in Section 2.2.

We define weak conjugate prior distributions over the model parameters $\boldsymbol{\theta}$ and θ^{act} using a symmetric Dirichlet distribution with hyperparameter $\boldsymbol{\alpha^{dir}} = (\alpha^{dir}, \ldots, \alpha^{dir})$ and Beta distribution with hyperparameters α^{act} and β^{act} respectively.

The following equations summarize the probability distributions of the model parameters and latent variables:

$$P(I_n|\boldsymbol{\theta}, Z_n^{act}) = \begin{cases} 0 & ; Z_n^{act} = 0 \\ \text{Cat}(I_n|\boldsymbol{\theta}) & ; Z_n^{act} = 1 \end{cases}, \tag{2.5}$$

$$P(Z_n^{act}|\theta^{act}) = \text{Bern}(Z_n^{act}|\theta^{act}), \tag{2.6}$$

$$P(\boldsymbol{\theta}) = \text{Dir}(\boldsymbol{\alpha^{dir}}), \tag{2.7}$$

$$P(\theta^{act}) = \text{Beta}(\alpha^{act}, \beta^{act}). \tag{2.8}$$

We use explicit modelling of the noise through the noise indicator Z_n^{act} and Beta distributed noise parameter θ^{act}. Note that marginal univariate distribution for a single component of a Dirichlet distributed vector is the Beta distribution. Hence including the noise parameter in the vector $\boldsymbol{\theta}$ and treating it as another transcript would result in an equivalent model. We depict the simplified model in Figure 2.2(b). This alternative definition is used in the Variational Bayes inference presented in Chapter 4.

2.2 Likelihood of read observation

In this section, we focus on estimating the likelihood of observing a read, given that we know a transcript of its origin m, $P(r_n|I_n = m)$. As described in Section 2.1.2, this likelihood can be viewed as the joint product of the likelihood of selecting a specific fragment and the likelihood of sequencing the actual read:

$$P(fragment|transcript)P(read|fragment).$$

A particular focus is given to non-uniform read distribution biases in contrast with uniform read distribution biases, which affect the distribution of fragments or $P(fragment|transcript)$.

Every read can be characterized by its position within transcript p, the strand to which it aligns s and its sequence r_n. As we show later, the only viable positions worth considering are those where the read actually aligns to the transcriptome, however for the moment p can be any arbitrary position within a transcript.

2.2.1 Single-end vs. paired-end reads

Depending on the type of sequencing used, the data consists of either single-end reads or paired-end reads. In the first case, only one of the ends of a fragment is sequenced. In the second case, each read has two mates, which are the result of sequencing both ends of a fragment. We can look at the pair of mates as one long read with unknown sequence in the middle that provides us with some extra information. Firstly, the alignment of the paired read will be more selective, as both ends have to come from the same transcript and secondly, it will provide us with information about the fragment's length.

Single-end reads In the case of single-end reads, the only information that we know about the fragment of a read's origin is the position of one of its ends and the lower bound on its length. The read was sequenced from one of fragment's ends, thus the starting position of a read p determines one end of a fragment. We also know that the transcript is at least as long as the read, hence the lower bound on fragment's length.

Given that we determined the read's transcript of origin $I_n = m$, we model the likelihood of the observation as a joint likelihood of selecting a strand s, choosing a fragment end at position p and the likelihood of read sequence generation, given the reference sequence seq_{mps},

$$P(r_n|I_n = m) = P(s|m)P(p|m)P(r_n|seq_{mps}). \tag{2.9}$$

Paired-end reads The paired-end reads cover both ends of fragments, which are in most cases length-selected before sequencing, thus providing a relatively narrow distribution of possible lengths. This distribution can be known beforehand and be part of the RNA-seq data description, or it can be inferred from the

data itself from uniquely aligned reads. Given the unique mappings of reads, the distances of their further ends are the lengths of the sequenced fragments.

We will denote r_n as being a pair of variables describing the individual reads, $r_n = (r_n^{(1)}, r_n^{(2)})$. Length of a fragment, denoted by l, is the distance of the furthest ends of mates in terms of transcript coordinates. For a given transcript $I_n = m$, the probability of observing paired reads $(r_n^{(1)}, r_n^{(2)})$ is determined by the probability of the read being sequenced from a specific strand s at a specific position p with a specific insert length l and the probability of reporting the reads after sequencing the reference sequences $(seq_{mlps}^{(1)}, seq_{mlps}^{(2)})$,

$$P(r_n^{(1)}, r_n^{(2)}|I_n = m) = P(s|m)P(l|m)P(p|l,m)P(r_n^{(1)}|seq_{mlps}^{(1)})P(r_n^{(2)}|seq_{mlps}^{(2)}) \ . \tag{2.10}$$

2.2.2 Strandedness

Current sequencing protocols are either strand specific or un-stranded. In the first case a specific strand is sequenced first and all reads will originate from that strand. We can say that the probability of observing a read aligning to the correct strand is always one

$$P(s = 1|m) = 1 \ . \tag{2.11}$$

A small fraction of alignments will match also to the incorrect strand, however the likelihood of this alignment being correct is difficult to evaluate. For this reason it is preferable to filter these alignments before the estimation, during the alignment stage.

On the other hand, un-stranded protocols can produce reads aligning to either strand with equal probability and thus the probability of observing a read from either strand is equal:

$$P(s = 1|m) = P(s = 0|m) = 0.5 \ . \tag{2.12}$$

As long as alignments to the incorrect strand are filtered for strand specific protocols, the likelihood of observing a read from a specific strand $P(s|m)$ does not play a role in our model. Even though we include this term in equations 2.9 and 2.10 for completeness, we leave it out from further calculations.

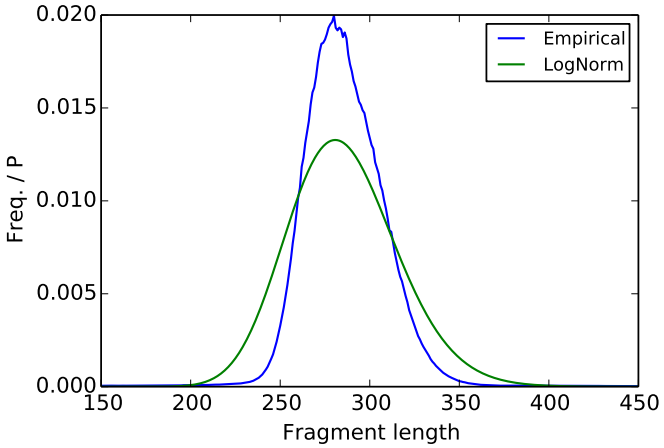


Figure 2.3: **Empirical fragment length distribution and approximation through Log-Normal distribution.**

2.2.3 Fragment length distribution

Because of the fragmentation protocol and fragment length filtering applied in sequencing protocols, the variance of lengths of sequenced fragments is small. The read fragment length distribution plays an important role in deconvolution of RNA-seq read alignments. For example, a read pair aligning to two transcripts that differ in sequence length between the two ends will have much higher probability of alignment to a transcript for which the ends are separated by a distance that is more likely according to the fragment length distribution. Despite the fragmentation process and filtering being universal for all transcripts, the maximal length of a fragment is limited by the maximal length of a transcript, hence we use the term $P(l|m)$ for the probability of observing a fragment of a specific length.

We approximate the probability distribution of fragment lengths by a Log-Normal distribution. An example of the empirical distribution of fragment lengths based on uniquely mapping reads and Log-Normal approximation is presented in Figure 2.3. For every transcript, we simply assume that the probability of fragment length in a range $[0, l_m]$ is proportional to a Log-Normal probability density, while for lengths above the transcript length l_m it is 0. We calculate the probability of fragment length l given the transcript m by normalising the Log-Normal probability with Log-Normal cumulative density. As the fragment lengths are integers, we calculate the cumulative density by summing over all possible fragment lengths. For a transcript of length l_m the probability of observing a

fragment of length l bases is

$$P(l|m) = \frac{\frac{1}{l\sigma_{len}\sqrt{2\pi}} \exp\left(-\frac{(\log l - \mu_{len})^2}{2\sigma_{len}^2}\right)}{\sum_{k=1}^{l_m} \frac{1}{k\sigma_{len}\sqrt{2\pi}} \exp\left(-\frac{(\log k - \mu_{len})^2}{2\sigma_{len}^2}\right)}, \tag{2.13}$$

with μ_{len} and σ_{len} being the parameters of Log-Normal distribution.

In some cases, the actual distribution, or at least its mean and variance are provided with the data itself. However, in most cases this information is missing and thus has to be inferred from the data. In a strict Bayesian approach, the parameters of the distribution would be treated as latent parameters of the model. In our case, we decided not to include these parameters in the model in order to simplify the inference process. The parameters are estimated empirically and the fact that there is great deal of data makes this approach feasible. Given the distribution of fragment lengths from $N^{(1)}$ uniquely mapping paired reads, we use Maximum Likelihood estimation for the Log-Normal distribution parameters:

$$\begin{aligned} \hat{\mu}_{len}^{(ML)} &= \frac{\sum_{i=1}^{N^{(1)}} \log l_i}{N^{(1)}}, \\ \hat{\sigma}_{len}^{2(ML)} &= \frac{\sum_{i=1}^{N^{(1)}} (\log l_i - \hat{\mu}_{len}^{(ML)})^2}{N^{(1)}}. \end{aligned} \tag{2.14}$$

2.2.4 Likelihood of read sequence observation

The reads reported by the sequencer are obtained by a process which is subject to experimental errors. The proportion of wrongly identified bases in high-throughput sequencing is relatively low and improving, however sequencing errors do occur. The majority of data generated by sequencers is provided in *Fastq* format that contains read sequences as well as base quality scores. The scores, in so called *Phred* format, provide an estimate of reliability for each base.

The quality Q_{Phred} encodes the probability of the base being incorrect, p_{err}:

$$\begin{aligned} Q_{Phred} &= -10 \log_{10} p_{err}, \\ p_{err} &= 10^{-Q_{Phred}/10}. \end{aligned} \tag{2.15}$$

We use this score to evaluate the probability of observing read sequence with

all its base mismatches given the underlying sequence of the reference transcriptome seq_{mlps}. The coordinates m, l, p, s encode the transcript, fragment length, fragment start position and strand, respectively. For a read of length l_r, the probability of observing its sequence is calculated as

$$P(r_n^{(1)}|seq_{mlps}) = \prod_{i=1}^{l_r} \Big(\delta(r_{n,i}^{(1)} = seq_{mlps,i})(1 - p_{err,i}) + \delta(r_{n,i}^{(1)} \neq seq_{mlps,i}) p_{err,i} \Big), \tag{2.16}$$

and equivalently for second paired read or for single-end case.

Accounting for the probability of read sequence generation enables inclusion of alignments with base mismatches in a sound way. Mismatches that are due to low sequencing quality have lesser effect on the likelihood than mismatches with high quality score. Hence alignment with extra mismatch on a base with high quality score will have much lower likelihood than alignment without the mismatch.

There is an exception in cases where a read covers a Single Nucleotide Polymorphism (SNP), in which case the base will not match the reference despite a high quality score. Again, a more complex model of the data could model the likelihood of such a scenario given all the reads. However, SNP detection and correction of the reference can be easier dealt with in the data processing stage using tools specifically designed for the task, such as SOAPsnp or GATK (Li et al., 2009a; DePristo et al., 2011).

2.2.5 Read distribution and fragmentation bias

The probability of sequencing a given position is in general given by

$$P(p|I_n = m, l) = \frac{b_m(p, l)}{\sum_{q=1}^{l_m - l + 1} b_m(q, l)}, \tag{2.17}$$

where $b_m(p, l)$ denotes the bias in sequencing a fragment of length l from a particular position p of a transcript m. For some data, it is viable to assume a uniform read distribution. In such case the bias $b_m(p, l)$ is constant which reduces the probability of sequencing position p to

$$P(p|m, l) = 1/(l_m - l + 1) \tag{2.18}$$

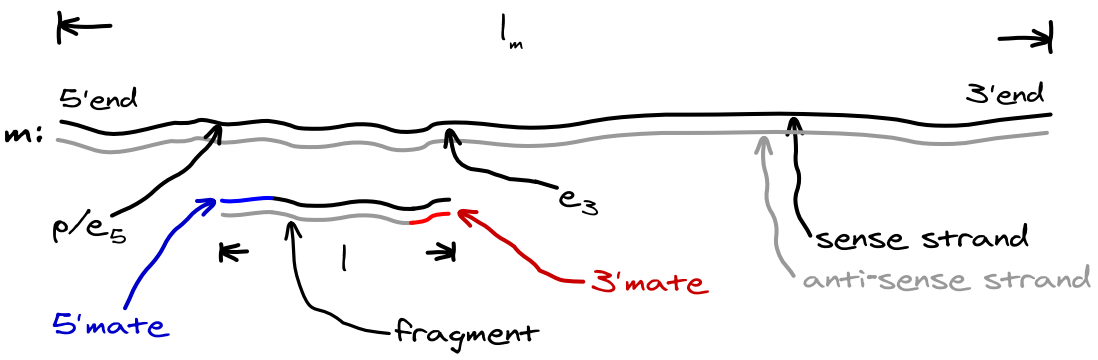


Figure 2.4: **Diagram of a sequenced fragment.** In most protocols the paired-end read's mates originate from ends of a fragment, with 3' mate being from the complementary reference strand.

depending only on transcript and fragment lengths. In the case of single end reads, the fragment length l is replaced by read length l_r.

The uniform read distribution assumption is in many cases incorrect. It has previously been stated by many studies that depending on the sequencing protocol and the properties of the transcriptome, the read distribution can be far from uniform (Howard and Heber, 2010; Wu et al., 2011; Li et al., 2010b; Roberts et al., 2011). The non-uniformities can be caused by sequence properties such as GC content, fragmentation bias as well as inclination to sequence specific regions of transcript molecules.

In order to account for the non-uniform read distribution, we applied a bias correction model introduced by Roberts et al. (2011). We selected this model as it seemed to be the most comprehensive and provided good empirical results. It divides the bias into two separate parts: the sequence specific bias and a position specific bias.

Given the properties of our generative model of RNA-seq data, it is easily extendible to incorporate the bias model proposed by Roberts et al. (2011). Assuming the parameters of the model are known, the bias model is used in form of a correction in the read alignment likelihood calculation. Similarly, any other read distribution model, which can be evaluated in terms of the likelihood of sequencing a position given transcript, fragment length and strand, $P(p|mls)$, could be used instead.

On the other hand, the bias model parameters are usually unknown and have to be inferred from the data. Inference of the bias model parameters with the data

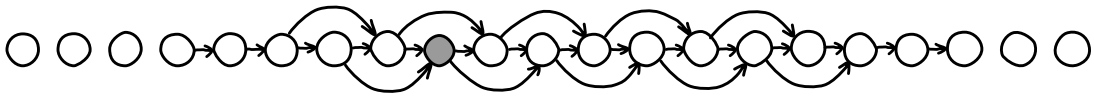


Figure 2.5: **Variable length Markov model for sequence specific fragmentation bias.** The nodes represent bases around the fragment start site, with the grey node being the first base of a fragment. Arrows mark conditional dependents of following bases.

is difficult and computationally expensive. We follow the approach of Roberts et al. (2011) in learning the bias model parameters *a priori* using an initial coarse estimate of the expression levels and fixing the bias model for the remaining inference.

Non-uniform read distribution bias model

In this section, we describe the implementation of a read distribution bias model which was originally presented by Roberts et al. (2011). Under this model, the bias for a given position of a transcript m and fragment of length l is the product of individual biases of both paired mates,

$$b_m(p,l) = b_m^{s,5}(e_5)b_m^{s,3}(e_3)b_m^{p,5}(e_5)b_m^{p,3}(e_3), \tag{2.19}$$

where $b_m^{s,5}(e_5)$ and $b_m^{s,3}(e_3)$ are the sequence specific biases for 5' and 3' ends of the fragment, respectively, and $b_m^{p,5}(e_5)$ and $b_m^{p,3}(e_3)$ are the corresponding positional biases.

Sequence specific bias We use variable length Markov models to capture the sequence specific bias for each end. The structure of the models, presented in Figure 2.5, is the same as that of Roberts et al. (2011). The bias of one mate is given as a product of probabilities of individual bases,

$$b_m^{s,5}(e_5) = \prod_{n=1}^{21} \frac{\psi_{n,\pi_n}^{5,R}}{\psi_{n,\pi_n}^{5,U}}, \tag{2.20}$$

based on 21 probabilities ψ_{n,π_n}^5 from 8 bases before and 12 bases after the read starting position. Here $\psi^{5,R}$ refers to the biased and $\psi^{5,U}$ to a uniform model, n is a node or a position, π_n are the parents of node n and ψ_{n,π_n}^5 is the probability of base X at node (or position) n given the bases observed on parent nodes π_n.

The model has 744 parameters in all, with each node having 0, 1 or 2 parents as in the model of Roberts et al. (2011).

The parameters are estimated from empirical frequencies using reads with a single alignment. As the transcripts with high expression produce more reads, we weight every read by the inverse expression of its transcript in order to avoid overfitting to the biases of highly expressed transcripts. For a read r aligning to transcript m we increase appropriate probabilities $\psi^{5,R}$ by $1/\theta_m$, where θ_m is an initial coarse expression, estimated with uniform read distribution model beforehand. In the contrasting uniform model for all $K = l_m - l + 1$ possible positions of fragment of length l, the appropriate probabilities $\psi^{5,U}$ are increased by $\frac{1}{\theta_m K}$. The model for 3' end, $b_m^{s,3}(e_3)$, is implemented similarly.

Position specific bias In addition to the sequence specific bias, there is a model for positional bias within the transcript:

$$b_m^{p,5}(e_5) = \frac{\omega^R_{l_m, e_5/l_m}}{\omega^U_{l_m, e_5/l_m}}, \tag{2.21}$$

where $\omega_{l,p}$ is the probability for starting position within transcript of length l_m on position p. The probabilities are modelled within 5 transcript length bins and 20 bins of relative position. The probabilities are again estimated from empirical frequencies of reads with single alignments weighted by the inverse expression $1/\theta$.

2.2.6 Effective length computation

We have introduced the concept of effective length in Section 1.4.2. It is the measure of the amount of reads that can be produced from a specific transcript molecule, which depends on the transcript length, fragment size distribution and read distribution non-uniformities. As our model infers expression in terms of θ, the relative proportion of transcript fragments, effective lengths are necessary for converting θ into RPKM or a measure of molecules instead of fragments.

For single-end reads the fragment length distribution is unknown, unless provided with the sequencing data. In such case, we can use the read length as a lower bound on the fragment length and approximate the effective length of a transcript of length l_m by the number of positions from which a read could have

been sequenced:

$$l_m^{(eff)} = l_m - l_r + 1, \tag{2.22}$$

where l_r is the read length.

For paired-end data the distribution of fragment sizes can be estimated empirically based on uniquely mapping reads. In such case the effective length can be estimated as the sum of all possible positions for all fragment lengths weighted by the likelihood of the fragment length,

$$l_m^{(eff)} = \sum_{l_f=1}^{l_m} P(l_f|m)(l_m - l_f + 1). \tag{2.23}$$

This expression assumes uniform distribution of fragments along transcripts as the term $(l_m - l_f + 1)$ counts every possible position of the fragment of length l_f equally.

Finally, in cases with known or empirically estimated read distribution bias model, we can extend Equation 2.23 by weighting each position with a bias weight. The effective length is then estimated as

$$l_m^{(eff)} = \sum_{l_f=1}^{l_m} P(l_f|m) \sum_{p=1}^{l_m - l_f + 1} b_m(p, l_f), \tag{2.24}$$

where $b_m(p, l_f)$ is the bias weight of position p within transcript m defined by Equation 2.19.

2.2.7 Probability of read being generated by noise

In the previous sections we covered the evaluation of the likelihood of a read originating from an actual transcript. In our model, we want to allow for some of the reads to be discarded and not assigned to any transcript. These could be for example either low quality reads or reads with alignments that are too ambiguous. For this purpose, we allow some of the reads to be assigned to an artificial *noise* transcript. Thus we have to define the likelihood of observing a read originating from noise, $P(r_n|I_n = 0)$ or simple $P(r_n|noise)$.

We again resort to an empirical approach in which we use the likelihoods of a read's alignments to true transcripts, $P(r_n|I_n > 0)$. We assume that the aligner reported all mappings of the read with up to X base errors. The noise transcript

is then modeled as a hypothetical mapping with $K > X$ base mismatches, for a fixed constant K. In other words, we expect that increasing the aligner's limit X might yield additional mapping to an unknown transcript, hence the read is assigned to noise in such case.

The $P(r_n|noise)$ is calculated by penalising the least probable alignment with extra base errors. Taking an alignment to a transcript with the lowest probability, $\breve{m} = \underset{m}{\operatorname{argmin}}\, P(r_n|m)$, we calculate the probability of the noise alignment using the same position, fragment length and strand, while adding more base mismatches to the observed sequence,

$$P(r_n|noise) = P(s|\breve{m})P(l|\breve{m})P(p|l,\breve{m})P(r_n^{(1)K}|seq^{(1)}_{\breve{m}lps})P(r_n^{(2)K}|seq^{(2)}_{\breve{m}lps}), \quad (2.25)$$

where $r_n^{(1)K}$ denotes the sequence of first mate of paired end read r_n with K total base errors. The additional base mismatches are added towards the end of the read sequence as those bases usually have the lowest quality score anyway.

2.3 Inference via Markov chain Monte Carlo

Section 2.1 detailed the probabilistic model for an RNA-seq dataset. Applying Bayes' theorem to the joint likelihood (Equation 2.4), yields the posterior distribution over model parameters and latent variables:

$$P(\boldsymbol{I}, \boldsymbol{Z^{act}}, \boldsymbol{\theta}, \theta^{act}|R) = \frac{P(\boldsymbol{\theta})P(\theta^{act})\prod_{n=1}^{N}(P(r_n|I_n)P(I_n|\boldsymbol{\theta}, Z_n^{act})P(Z_n^{act}|\theta^{act}))}{P(R)}. \quad (2.26)$$

The marginal likelihood of the data $P(R)$ is intractable, hence we have to resort to an approximate inference of the model parameters. A natural choice is to use a Markov chain Monte Carlo (MCMC) algorithm, which enables drawing samples from the posterior distribution without the need to evaluate the marginal likelihood. Thanks to the conjugacy of prior distributions over parameters, we can use Gibbs sampling and Collapsed Gibbs sampling approaches which are described in Sections 2.3.2 and 2.3.3 below.

2.3.1 Computing read observation likelihoods

Computationally efficient inference of this method is enabled by two factors. Firstly, as can be seen from the graphical representation in Figure 2.2(a), the

likelihood of observing a read r_n is conditionally independent of model parameters given read's assignment I_n. Secondly and most importantly, the likelihood of read observation has to be considered only for transcripts and positions for which an alignment has been found.

Every read could have possibly originated from any transcript, and as noted earlier, every read could have possibly originated from any position of a transcript. The fact that a position has not been reported as a read's alignment means that the read differs by at least $k + 1$ bases, where k is the number of allowed mismatches. Therefore, the probability of a read being from a transcript with no reported alignments or from a position which has not been reported is vanishingly small in comparison with the probability of the read originating from one of the reported alignments. Hence we only evaluate the read observation likelihood, $P(r_n|I_n = m)$, for transcripts to which the read aligns, and only consider positions, lengths and strands (p, l, s) of valid alignments.

Conditional independence of the read observation likelihood allows pre-computing of all the likelihoods before applying the MCMC algorithm. Given a reference sequence and all the alignments (which contain read sequence, transcript identifier, strand, position fragment length in case of paired reads) a sparse matrix is computed. The matrix has one row per transcript and one column per read and a position $[m, n]$ contains the likelihood of read r_n originating from transcript m. In an exceptional case when a read has more valid alignments to a transcript, it would be a sum of observation likelihoods of all those alignments.

2.3.2 Gibbs Sampling

In Section 1.5.5 we introduced the Gibbs Sampling algorithm (Geman and Geman, 1984). The main principle of the algorithm is that sampling from a posterior distribution of all parameters is achieved via iterative sampling from conditional distributions of individual parameter vectors. To sample the $(t + 1)$-th sample from the posterior distribution, we use parameters sampled in the previous iteration and sample from following conditional distributions:

$$\begin{aligned}
\boldsymbol{Z}^{\boldsymbol{act}}{}_{(t+1)} &\sim P(\boldsymbol{Z}^{\boldsymbol{act}}{}_{(t+1)}|\boldsymbol{I}_{(t)}, \boldsymbol{\theta}_{(t)}, \theta^{act}{}_{(t)}, R),\\
\boldsymbol{I}_{(t+1)} &\sim P(\boldsymbol{I}_{(t+1)} \quad |\boldsymbol{Z}^{\boldsymbol{act}}{}_{(t+1)}, \boldsymbol{\theta}_{(t)}, \theta^{act}{}_{(t)}, R),\\
\boldsymbol{\theta}_{(t+1)} &\sim P(\boldsymbol{\theta}_{(t+1)} \quad |\boldsymbol{I}_{(t+1)}, \boldsymbol{Z}^{\boldsymbol{act}}{}_{(t+1)}, \theta^{act}{}_{(t)}, R),\\
\theta^{act}{}_{(t+1)} &\sim P(\theta^{act}{}_{(t+1)} \quad |\boldsymbol{I}_{(t+1)}, \boldsymbol{Z}^{\boldsymbol{act}}{}_{(t+1)}, \boldsymbol{\theta}_{(t+1)}, R).
\end{aligned} \tag{2.27}$$

Before we evaluate the conditional distributions, we further simplify the inference by marginalizing out the latent variable $\boldsymbol{Z^{act}}$. We are mostly interested in parameter $\boldsymbol{\theta}$, thus we do not need to sample $\boldsymbol{Z^{act}}$ and, as we will show, integrating out $\boldsymbol{Z^{act}}$ will not make sampling from conditional distributions of other parameters intractable. The algorithm samples from the following posterior distribution:

$$\begin{aligned} P(\boldsymbol{I}, \boldsymbol{\theta}, \theta^{act}|R) &= \sum_{\boldsymbol{Z^{act}}} P(\boldsymbol{I}, \boldsymbol{\theta}, \theta^{act}|\boldsymbol{Z^{act}}, R) P(\boldsymbol{Z^{act}}) \\ &\propto P(\boldsymbol{\theta}) P(\theta^{act}) \prod_{n; I_n \neq 0} \left(P(r_n|I_n) \mathrm{Cat}(I_n|\boldsymbol{\theta}) \theta^{act} \right) \\ &\quad \prod_{n; I_n = 0} \left(P(r_n|noise)(1-\theta^{act}) \right). \end{aligned} \tag{2.28}$$

A detailed derivation of this formula can be viewed in Section A.1 of the appendix.

We derive the conditional distributions from Equation 2.28 using the basic probabilistic rule,

$$P(\alpha|\beta, R) = \frac{P(\alpha, \beta|R)}{P(\beta|R)} \propto P(\alpha, \beta|R)_{\beta}, \tag{2.29}$$

which states that the conditional distribution of α given all other parameters is proportional to the joint likelihood taken as a function of α.

The derived conditional distributions are

$$\begin{aligned} P(I_n|\boldsymbol{\theta}, \theta^{act}, R) &= \mathrm{Cat}(I_n|\boldsymbol{\phi_n}), \\ \phi_{n0} &= P(r_n|noise)(1-\theta^{act})/Z_n^{(\phi)}, \\ m \neq 0; \phi_{nm} &= P(r_n|m)\theta_m \theta^{act}/Z_n^{(\phi)}, \\ Z_n^{(\phi)} &= P(r_n|noise)(1-\theta^{act}) + \textstyle\sum_{m=1}^{M} P(r_n|m)\theta_m \theta^{act}, \\ P(\boldsymbol{\theta}|\boldsymbol{I}, \theta^{act}, R) &= \mathrm{Dir}(\boldsymbol{\theta}|(\alpha^{dir} + C_1, \ldots, \alpha^{dir} + C_M)), \\ P(\theta^{act}|\boldsymbol{I}, \boldsymbol{\theta}, R) &= \mathrm{Beta}(\theta^{act}|\alpha^{act} + N - C_0, \beta^{act} + C_0), \\ C_m &= \textstyle\sum_{n=1}^{N} \delta(I_n = m). \end{aligned} \tag{2.30}$$

These are all in a form of standard probability distributions, thus making it straightforward to sample the parameters. Also note that $\boldsymbol{\theta}$ and θ^{act} are conditionally independent given $\boldsymbol{I}$ and thus only the counts of read assignments, $\boldsymbol{C}$, are necessary when sampling these variables.

2.3.3 Collapsed Gibbs Sampling

Collapsing is the process of marginalizing out latent variables, which improves the convergence speed of a Gibbs sampler (Liu, 1994). Griffiths and Steyvers (2004) introduced Collapsed Gibbs Sampling for Latent Dirichlet Allocation, a hierarchical probabilistic model similar to our generative model of an RNA-seq experiment. Using a similar approach, we marginalize θ^{act} and $\boldsymbol{\theta}$ and sample only the assignments of reads to transcripts $\boldsymbol{I}$. However, $\boldsymbol{\theta}$ is the main parameter of interest, thus we will have to sample $\boldsymbol{\theta}$ from the conditional distribution given $\boldsymbol{I}$ (see Equation 2.30), when generating the output samples.

Following from Equation 2.28, we first integrate the posterior over all possible values of θ^{act},

$$\begin{aligned} P(\boldsymbol{I}, \boldsymbol{\theta}|R) = & \int_0^1 \mathrm{d}\theta^{act} P(\boldsymbol{I}, \boldsymbol{\theta}|\theta^{act}, R) P(\theta^{act}) \\ & \propto P(\boldsymbol{\theta}) \prod_{n;I_n \neq 0} (P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta})) \prod_{n;I_n=0} P(r_n|noise) \\ & \quad \Gamma(\alpha^{act} + C_+)\Gamma(\beta^{act} + C_0), \end{aligned} \tag{2.31}$$

where $C_+ = \sum_{m=1}^{M} C_m = N - C_0$ and Γ is the Gamma function. Now we can further integrate over all values of $\boldsymbol{\theta}$ to obtain conditional posterior distribution of read assignments:

$$\begin{aligned} P(\boldsymbol{I}|R) = & \int_{\boldsymbol{\theta}} \mathrm{d}\boldsymbol{\theta} P(\boldsymbol{I}|\boldsymbol{\theta}R) P(\boldsymbol{\theta}) \\ & \propto \prod_{n;I_n \neq 0} P(r_n|I_n) \prod_{n;I_n=0} P(r_n|noise) \frac{\prod_{m=1}^{M} \Gamma(\alpha^{dir} + C_m)}{\Gamma(M\alpha^{dir} + C_+)} \\ & \quad \Gamma(\alpha^{act} + C_+)\Gamma(\beta^{act} + C_0). \end{aligned} \tag{2.32}$$

As we integrated out $\boldsymbol{\theta}$ and θ^{act}, the individual assignments of reads, I_n, are no longer conditionally independent. This is not a problem though, as the principle of Gibbs sampler is in the sampling of individual parameters conditioned on the rest. Instead of considering the vector $\boldsymbol{I}$ as single parameter, we can look at each read individually and sample assignment for each read given the current state of the other reads, i.e. other assignments. From Equation 2.32, the assignment of

the n-th read conditioned on the rest follows a Categorical distribution:

$$\begin{aligned} P(I_n|I^{(-n)}, R) &= \mathrm{Cat}(I_n|\boldsymbol{\phi}^*_{\boldsymbol{n}}), \\ \phi^*_{n0} &= P(r_n|\text{noise})(\beta^{act} + C_0^{(-n)})/Z_n^{(\phi^*)}, \\ m \neq 0; \phi^*_{nm} &= P(r_n|m)(\alpha^{act} + C_+^{(-n)})\frac{(\alpha^{dir}+C_m^{(-n)})}{(M\alpha^{dir}+C_+^{(-n)})}/Z_n^{(\phi^*)}, \\ C_m^{(-n)} &= \textstyle\sum_{i\neq n}\delta(I_i = m), \\ C_+^{(-n)} &= \textstyle\sum_{i\neq n}\delta(I_i > 0)\ . \end{aligned} \tag{2.33}$$

Hence in order to sample a single MCMC sample from the posterior distribution, we iteratively sample a transcript assignment for every read. After re-assigning all the reads, we can sample $\boldsymbol{\theta}$ based on the read counts per transcript.

2.3.4 Convergence checking

The Markov chain Monte Carlo sampler is guaranteed to sample from the desired distribution if an infinite number of samples is produced. For practical applications, we want to approximate this by producing as few samples as necessary. Also, the initial samples are affected by random initialization of the variables. Hence, estimation and assessment of the samples' convergence is an important part of the MCMC sampling. We follow principles covered by Gilks et al. (1995) and Gelman et al. (2004).

Burn in

To overcome the bias of starting position it is natural to discard the first samples, which are referred to as burn-in. The usual choice when generating L samples is to consider the first half as burn-in and discard it. If the convergence of the resulting sequence is not satisfactory, L more samples are produced and all L initial samples are discarded. However, in our case, the process of doubling the number of samples is no longer feasible after a few iterations, due to the large number of parameters being inferred.

Thinning

When the number of generated samples is too large, keeping all the samples is infeasible. This is especially true in cases such as ours when the parameter being sampled, $\boldsymbol{\theta}$, can have more than hundred thousand individual scalar values.

Thinning is the process of sub-sampling the set of generated samples by keeping only every k-th sample.

Apart from saving space, thinning also has another advantage. Despite all samples being correctly sampled from the posterior distribution, consecutive draws generated by MCMC are usually correlated. By thinning the sequence of samples we also reduce the correlation of consecutive samples.

Multi-chain comparison

The most important part of surveying posterior samples is the convergence assessment. We do this by running multiple independent sampling algorithms, or chains. Each chain starts by random initialization of the variables and will eventually sample from the true posterior. By comparing distributions generated by individual chains, we can draw conclusions on their convergence. As all chains start with different parameters, the initial samples will be different. However, if the initial bias has been overcome and all chains are sampling from the posterior, the distribution of the samples will be similar.

To assess the overall convergence, we used the $\widehat{R}$ statistic described by Gelman et al. (2004). It is based on the comparison of within (W) and between (B) sequence variance of K sequences each having L samples: $\theta_{k,l}$. Within sequence variance is the average sample variance for each sequence:

$$\begin{aligned} W &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{L-1}\sum_{l=1}^{L}(\theta_{k,l}-\bar{\theta}_{k,*})^2\right), \\ \bar{\theta}_{k,*} &= \frac{1}{L}\sum_{l=1}^{L}\theta_{k,l}. \end{aligned} \tag{2.34}$$

Between sequence variance is the variance of chain means given by

$$\begin{aligned} B &= \frac{L}{K-1}\sum_{k=1}^{K}(\bar{\theta}_{k,*}-\bar{\theta}_{*,*})^2, \\ \bar{\theta_{*,*}} &= \frac{1}{K}\sum_{k=1}^{K}\theta_{k,*}. \end{aligned} \tag{2.35}$$

Using these variances, the marginal posterior variance of the parameter can

be estimated by

$$\widehat{var}(\theta|R) = \frac{L-1}{L}W + \frac{1}{L}B. \tag{2.36}$$

The $\widehat{R}$ estimates the possible scale reduction of the marginal posterior variance

$$\widehat{R} = \sqrt{\frac{\widehat{var}(\theta|R)}{W}}. \tag{2.37}$$

As it was stated by Gelman et al. (2004), the limit of $\widehat{R}$ is 1 for $L \to \infty$, and thus values of all scalar parameters being close to 1 can be acceptable as a convergence criterion, with 1.1 being regarded as sufficiently close in most cases.

The $\widehat{R}$ estimate is intended for scalar variables that are normally distributed. We use marginal distributions of transcript expression θ_m and apply *logit* transformation. Using the transformed expression we calculate $\widehat{R}$ estimate for each transcript independently.

Iterative sample generation

A simple scheme involving the $\widehat{R}$ statistic for convergence checking relies on an iterative increase of the number of generated samples. The process is repeated until the convergence criterion is met.

At the beginning, $2 \times L$ samples are generated, L samples being discarded as burn-in and L samples used for assessing the convergence using the $\widehat{R}$ statistic. In case of a sufficiently small $\widehat{R}$, the L samples are considered as the true samples from the posterior distribution.

If, on the other hand, the convergence is not sufficient, the L samples are discarded. In this case, the $2 \times L$ samples are now considered burn-in and a new $2 \times L$ samples are generated. The process is repeated with the first half being always discarded and the amount of samples kept and used for convergence verification being doubled. As mentioned earlier, thinning can be applied if the amount of samples is unnecessarily large for further applications.

Estimating the effective number of independent samples

Gelman et al. (2004) provide a way to approximate the number of independent draws from the posterior distribution. Using the marginal posterior variance estimate $\widehat{var}(\theta|R)$ and between chain variance B, the number of effective independent

samples is

$$L^{(eff)} = KL\frac{\widehat{var}(\theta|R)}{B}. \tag{2.38}$$

We use this idea in order to provide a more efficient sampling scheme. The number of samples that are to be recorded and used, $L^{(r)}$ is usually predetermined. We use K independent chains and start by sampling L burn-in samples and L samples in order to generate first estimate of $\widehat{var}(\theta|R)$. While $\widehat{R}$ is greater than 1, the variance is expected to improve with producing more samples. This means that we can use current estimate of $\widehat{var}(\theta|R)$ in order to estimate how many draws from each chain have to be sampled in order to produce $L^{(r)}$ effective independent samples. Substituting $L^{(r)}$ into Equation 2.38 instead of $L^{(eff)}$:

$$L = \frac{L^{(r)}B}{K\widehat{var}(\theta|R)}, \tag{2.39}$$

L is the estimated number of samples that have to be sampled from each chain in order to produce $L^{(r)}$ effective independent draws from the posterior distribution.

This approach of deciding the number of necessary samples does not increase the burn-in and only scales the number of draws from each chain. This is desired in most cases, as the chains have converged, but only need to produce sufficient number of samples to cover the entire distribution.

2.4 Results and evaluation

We evaluate the proposed model and inference procedures using both artificial and real RNA-seq data. We apply our method to a simple 'toy' example generated from the model and to synthetic RNA-seq reads sampled based on the true sequencing process. The artificial data provide the advantage of known ground truth that enables exact assessment of the accuracy of our method. On the other hand, we analyse two real RNA-seq datasets that fully capture the complexity of the sequencing data, but lack means for full validation of the accuracy of our results.

The presented method and inference algorithms were implemented in the BitSeq application. BitSeq is implemented in `C++` as a standalone application consisting of individual programs for data preparation, expression inference and manipulation of the results. We have also created an `R` interface, which is part of the Bioconductor project (Gentleman et al., 2004) and enables the use of BitSeq

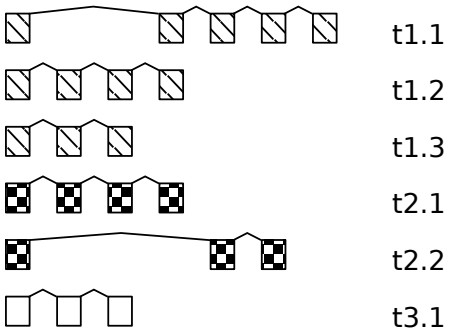


Figure 2.6: **Exon structure of toy transcriptome reference.** The reference consists of six transcripts from three different genes which are distinguished by three different fill patterns. Reads from vertically aligned exons of one gene are mapped to all transcripts containing those exons.

gene	transcript	length	μ	$C(D=1)$	θ
1	t1.1	5	6	30	0.33
1	t1.2	4	1	4	0.04
1	t1.3	3	4	12	0.13
2	t2.1	4	2	8	0.09
2	t2.2	3	7	21	0.23
3	t3.1	3	5	15	0.17

Table 2.1: **Pre-defined expression levels for toy reference data.** Length l depends on the number of exons, μ is the abundance of molecules, C is the read count for a given depth D and θ is the relative proportion of fragments.

from within the `R` environment. In the comparisons with alternative methods, we use the name BitSeq to refer to the model and inference procedure presented above.

2.4.1 Analysis of 'toy' example

We demonstrate the workings of the inference methods using a simple example with a toy reference dataset and perfectly uniform distribution. The reference structure is depicted in Figure 2.6. It consists of three genes with three, two and one transcripts respectively. The transcripts consist of exons of unit length which are shared by transcripts within one gene.

We assigned an absolute expression μ to every transcript which can be thought of as a number of molecules of that transcript. For a sequencing depth D, D reads are 'sequenced' from each exon of each molecule, hence the relative proportion of fragments $\boldsymbol{\theta}$ is constant. Table 2.1 shows the pre-defined values.

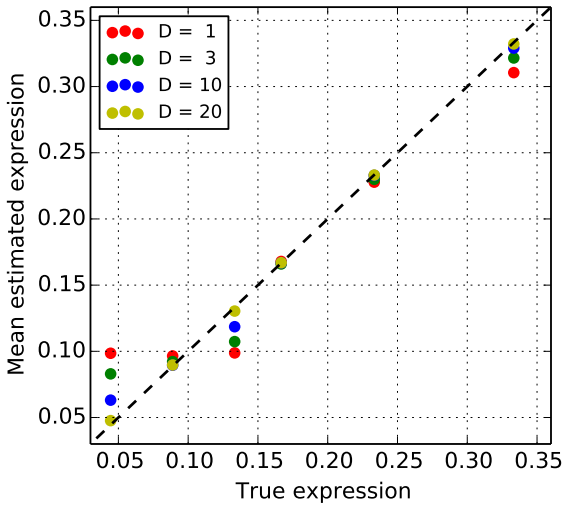


Figure 2.7: **Estimation accuracy on toy data for various sequencing depths.** The estimated mean expression is compared against the ground truth relative proportions of fragments.

Instead of sampling the reads, we directly generated the alignments and their probabilities based on transcript length. A read from a transcript of length l has probability $1/l$ of coming from that transcript. As an example, a read coming from the first exon of the transcript t1.1 has following alignment probabilities.

transcript	t1.1	t1.2	t1.3	t2.1	t2.2	t3.1
$P(read\|transcript)$	0.20	0.25	0.33	0	0	0

We generated alignments for depths $D = 1, 3, 10, 20$. For depth $D = 1$, 138 alignments of 90 reads were generated, while for higher depths these are just respective multiples. Subsequently, we used the MCMC algorithm to generate 1000 samples from the posterior distribution of $\boldsymbol{\theta}$.

The comparison of the mean estimated expression against ground truth from Table 2.1 is shown in Figure 2.7. The expression levels of transcripts of second and third genes are almost perfectly estimated even for the lowest depth $D = 1$. The transcripts of the first gene are harder to distinguish due to many exons being shared. Note that none of the reads generated from transcripts t1.2 and t1.3 have unique alignments, all are either shared between these two transcripts or with transcript t1.1. Despite this difficulty, due to correct alignment probability and perfectly uniform coverage, the inference algorithm provides accurate estimate of expression levels for high-enough sequencing depth.

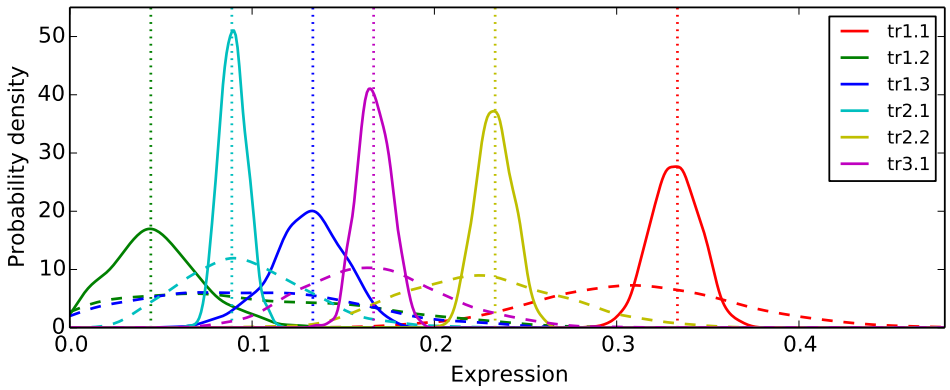


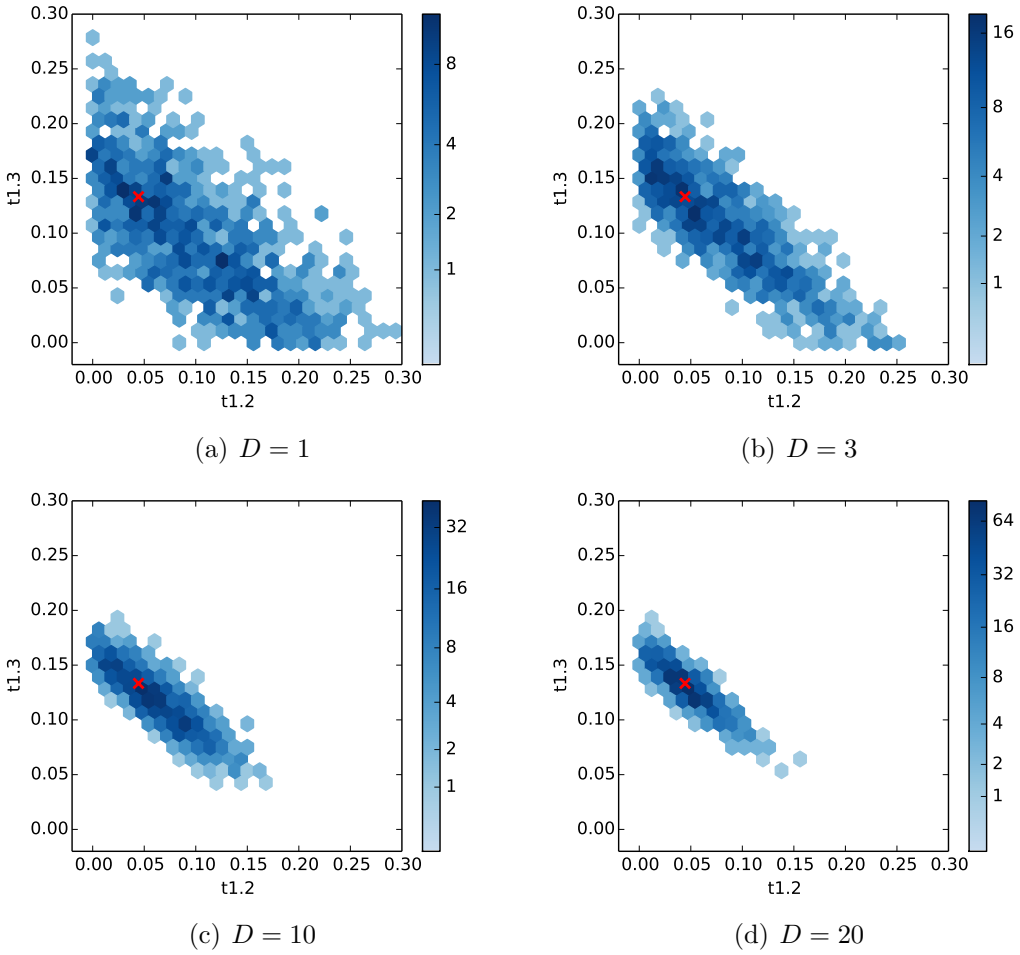
Figure 2.8: **Posterior probability densities of expression of six toy transcripts smoothed by kernel density estimation.** The solid lines represent results for data generated with depth $D = 20$, dashed lines represent data generated with depth $D = 1$. Vertical dotted lines mark ground truth expression for each transcript.

The posterior distributions of transcript expression levels inferred for depths 1 and 20 are shown in Figure 2.8. We smoothed the histogram of MCMC samples by kernel density estimation with Gaussian kernel. The increased number of reads due to higher depth clearly reduces the variance of the expression estimates.

We can further investigate the ambiguity of transcripts t1.2 and t1.3 and the effect of shared reads on their expression level estimates. Figure 2.9 shows four density plots of expression samples generated by the MCMC algorithm for four datasets with varying depth. The figures similarly show higher variance of expression of both transcripts for lower sequencing depth. Increasing sequencing depth lowers the uncertainty, hence variance decreases.

Most importantly, for all four depths clear anti-correlation of the expression levels can be observed. Higher expression of one of the transcripts implies lower expression of the other transcript. This is caused by ambiguous reads being alternatively assigned to one of the transcripts.

Analysis of the example dataset demonstrates the ability to estimate accurate expression levels for transcripts provided sufficient read coverage. In this example the reads were generated uniformly along the transcripts and the alignment probabilities correspond to the exact likelihood of read generating from an exon. While these 'perfect' conditions are never reached in real data analysis, we show in following experiments that they are not required to provide accurate expression estimates for most transcripts.



(a) $D = 1$ (b) $D = 3$

(c) $D = 10$ (d) $D = 20$

Figure 2.9: **Density plots of expression samples of transcripts t1.2 and t1.3 for varying depth levels of the data.** The expression is reported as θ, the relative proportion of fragments. The true expression level of both transcripts is marked with a red `x`.

2.4.2 Analysis of RNA-seq data from microRNA target identification study

We analysed RNA-seq data from a previously published microRNA target identification study by Xu et al. (2010). We used this data to examine transcript expression quantification using our probabilistic model as well as for the differential expression analysis method presented in Chapter 3. The dataset consists of two conditions, with two biological replicates each and multiple technical replicates.

We aligned the data to the UCSC NCBI37/hg19 knownGene transcriptome reference (Hsu et al., 2006; Meyer et al., 2013) using the Bowtie alignment

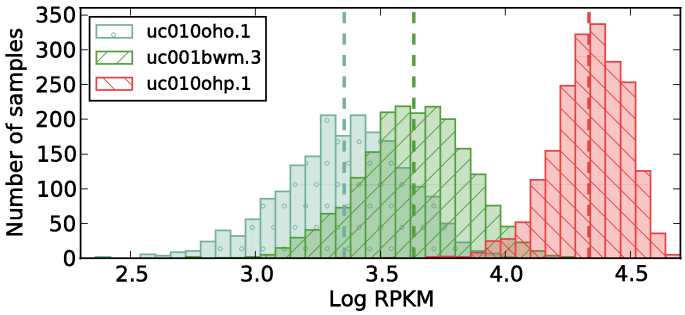


Figure 2.10: **Posterior distribution of expression levels of three transcripts of gene Q6ZMZ0.** The posterior distribution is represented in form of a histogram of expression samples converted into Log RPKM expression measure. The dashed lines mark the mean expression for each transcript.

tool (Langmead et al., 2009).

Within gene correlations of transcript expression

Here we focus on the expression estimates for three transcripts of a single gene Q6ZMZ0. Within the knownGene annotation, the gene has three different transcripts, uc010oho.1, uc001bwm.3 and uc010ohp.1. The inferred posterior distribution of expression converted into Log RPKM is plotted in Figure 2.10.

According to the observed reads and our inference procedure all three transcripts of the gene are present within the sample. However, the posterior distribution of transcripts shows anti-correlations between individual transcripts similar to those in the example shown in Figure 2.9. This is due to the ambiguity of reads aligning to multiple transcripts of a single gene. The pairwise density plots of posterior distributions of transcript expression are shown in Figure 2.11.

The expression estimates of transcript pairs uc011oho.1, uc010ohp.1 (Figure 2.11(a)) and uc001bwm.3, uc010ohp.1 (Figure 2.11(c)) are negatively correlated. This means that the model is unable to decide from which transcript some reads originated and the posterior distribution captures all viable assignments. When more reads are assigned to one of the transcripts its expression level increases and conversely, the expression level of the other transcript decreases.

The transcript sequence profile in Figure 2.12(b) clearly demonstrates the similarity of the transcripts that causes higher uncertainty when inferring the transcript expression levels. The transcripts share all but two constitutive sequences, the third exon and 3' untranslated region (UTR), which are specific

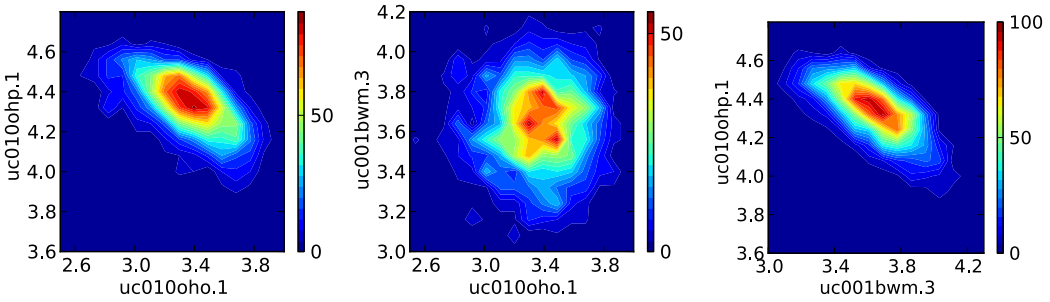


Figure 2.11: **Pairwise density plots of posterior distributions of transcript expression levels.** Each figure shows density plot of expression distribution for pair of transcripts of a single gene based on the samples produced by the MCMC inference algorithm. While transcript pairs in (a) and (c) show anti-correlation of estimated expression levels, there is no observable correlation between expression levels of transcripts in (b).

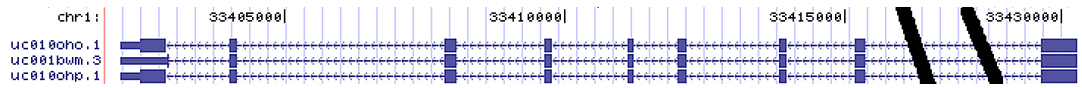
to the first and second transcripts, respectively. Only the first two transcripts which differ in both exon and UTR do not clearly show correlation of estimated expression levels.

The anti-correlations of the expression samples do not have biological significance. They can only indicate similarity of transcripts in the reference annotation and the fact that the probabilistic model cannot distinguish between transcripts. This information can be further used in the downstream analysis. Firstly, transcripts that are correlated have increased variance of expression levels which can avoid false positive differential expression calls. Secondly, highly correlated transcripts can be treated jointly, decreasing the variance of the joint transcript and enabling differential expression comparison of the joint transcript.

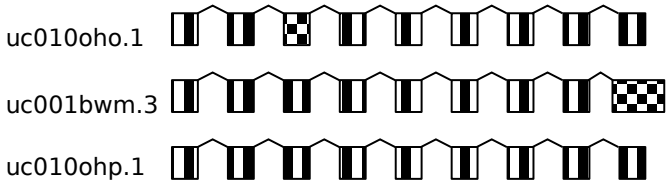
Transcript expression level variances

Correct estimation of deviations in expression levels is important for the detection of DE transcripts and genes. We use the dataset published by Xu et al. (2010) to examine the variances present in RNA-seq expression quantification experiments. As the dataset contains both technical and biological replicates we can compare the expression level deviations caused by technical noise and intrinsic fluctuations of transcript abundances.

Figure 2.13 shows the standard deviation of transcript expression level posterior MCMC samples as a function of the mean expression level of the transcript.



(a) Transcript sequence profile.



(b) Splice variant model.

Figure 2.12: **Exon model of transcripts of gene Q6ZMZ0.** (a) transcript sequence profile obtained from the UCSC genome browser (Kuhn et al., 2013). In this annotation, transcript uc001bwm.3 has different 3' untranslated region and transcript uc010oho.1 has extra nucleotides at the end of second exon. As the second change cannot be distinguished in the UCSC genome browser diagram, we provide schematic splice variant model highlighting the differences (b).

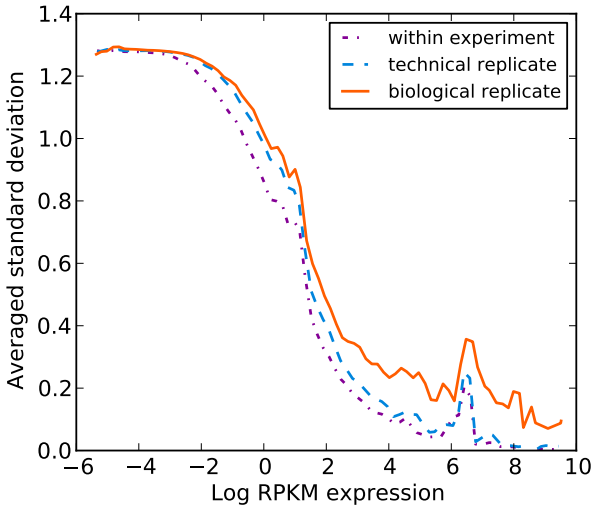


Figure 2.13: **Comparison of standard deviation of posterior samples within single run and combined data of technical replicates and biological replicates.** The plot shows mean Log RPKM expression of transcripts on the x-axis and averaged standard deviation of the Log RPKM expression on the y-axis. The standard deviation is a sliding average over groups of transcripts with similar expression in order to highlight its dependents on the expression.

The MCMC samples were converted into Log RPKM expression measure, which we use as an input to the DE analysis method presented in Chapter 3. We compare the standard deviation for samples within one experiment, between two technical replicates and between two biological replicates. In order to calculate the standard deviation between replicates we took the square root of variance which was estimated by computing the root mean square distance between samples. Plotted values are averaged for a sliding window of similarly expressed transcripts.

The MCMC sample variation captures the intrinsic estimation variance in the "within-experiment" case and includes both the random sampling noise and ambiguity of transcript sequences. The technical variance includes a contribution due to re-sequencing of a single biological sample while the biological variance includes a contribution due to natural fluctuations of abundances of transcripts within condition.

Similarly to previous results of Anders and Huber (2010), we observe significant biological variation within conditions. With higher expression the variance of the expression level estimation decreases, as can be expected due to greater evidence in form of increased coverage of reads. At high expression levels the variance associated with technical replicates approaches the level of the within-experiment variance. On the other hand, the biological variance becomes relatively more significant for transcripts with high expression level. Without consideration of the biological differences, the high confidence in expression level estimates of these transcripts will lead to false differential expression calls.

It can be further observed that the within-experiment variance has a significant contribution to the replicate variance (technical and biological) at lower expression levels. Therefore the intrinsic variance due to mapping ambiguity and limited read depth, as estimated by our MCMC expression estimation procedure, will provide useful information for assessing replicate variance in this low expression regime.

2.4.3 Analysis of RNA-seq data from the ENCODE project

We also analysed RNA-seq reads sequenced and published by the ENCODE consortium (Djebali et al., 2012), consisting of eight technical replicates, downloaded from the Short Read Archive (NCBI, 2010), accession number SRX159824. The data was produced by sequencing human embryonic stem cell line (H1-hESC)

with Illumina Genome Analyser, using a paired-end sequencing protocol. Here we focus on results for one replicate (SRR521478) containing 24.3 million read pairs. We mapped the reads using Bowtie (Langmead et al., 2009) to the Gencode reference (Harrow et al., 2012), producing 62M alignments of 14.5M reads. The Gencode reference is a curated set of human genes and transcripts containing 20720 genes with 94917 transcripts.

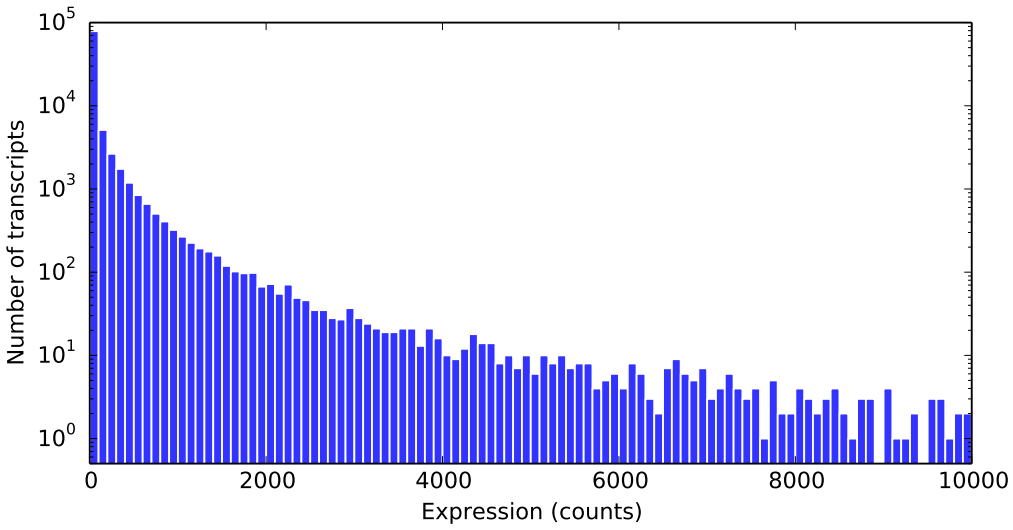
Transcript and gene expression levels

We analysed the reads using our probabilistic model with non-uniform read distribution correction. The resulting mean expression levels of transcripts are presented in Figure 2.14, showing large expression range for the top highly expressed transcripts. While our model infers expression in terms of $\boldsymbol{\theta}$, the relative proportion for fragments, it is more natural to look at expression in terms of the number of observed reads. We multiplied $\boldsymbol{\theta}$ by the total number of mapped reads in order to produce the estimated counts per transcript.

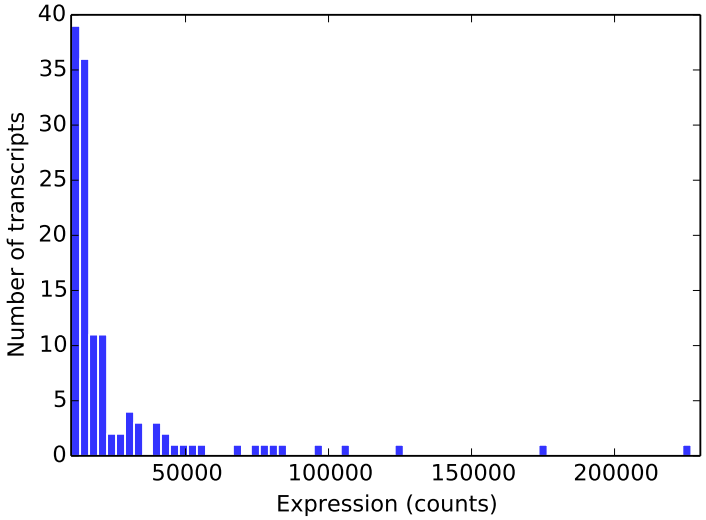
We further examined the variance of the estimated expression levels by looking at the sample variance of inferred marginal distributions of transcript expression. Figure 2.15 shows the relationship between mean expression level and variance. Certain number of transcripts exhibit only the Poisson variance of random sampling, in which case the variance is equal to the mean of the distribution. The rest of transcripts have much higher variance of transcript expression due to ambiguity of transcripts.

Except for transcript expression levels, many researchers are primarily interested in the gene expression, or the abundance of entire loci. We can simply calculate the gene expression by summing expression levels of individual transcripts of each gene. Figure 2.16 shows a histograms of gene expression measured in estimated counts. The gene counts cannot be directly transformed into the abundance of molecules as the effective length of a gene is not well defined. Nevertheless, counts are natural way of looking at the results of sequencing process.

As we convert the entire posterior distributions of transcript expression levels into gene expression levels, we can again investigate the variance of marginal distributions of gene expression levels. The mean-variance relationship of gene expression levels is shown in Figure 2.17. In accordance with our expectations, the gene expression variance is lower than the transcript expression variance. This further confirms that the transcript expression variance is inflated by the

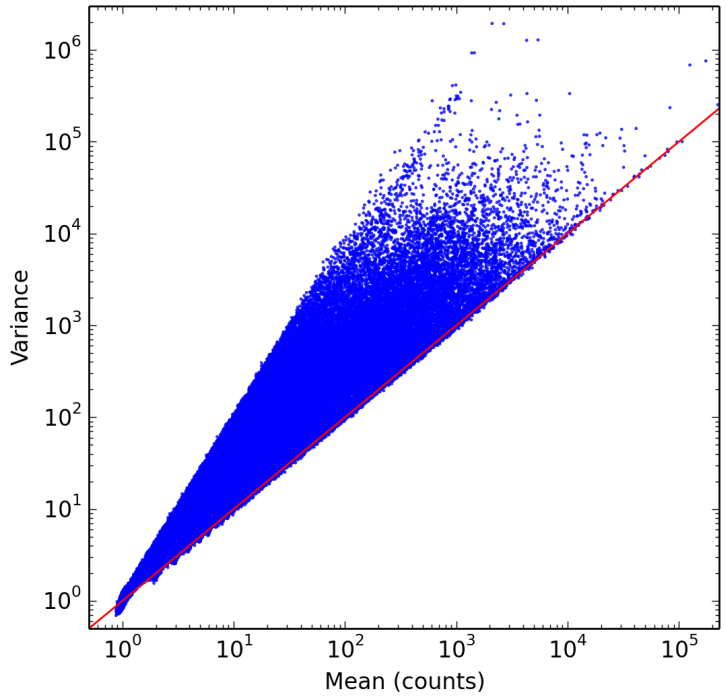


(a)

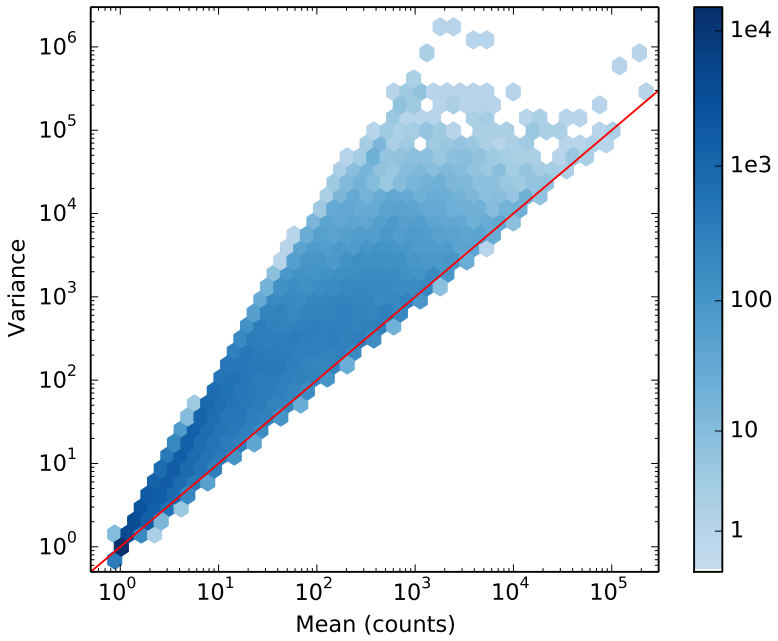


(b)

Figure 2.14: **Histograms of mean transcript expression levels in counts.** The entire range is split into two histograms: (a) majority of transcripts have estimated read count below 10^4, we use logarithmic y-axis, (b) the top 127 expressed transcripts are spread over a large expression range.

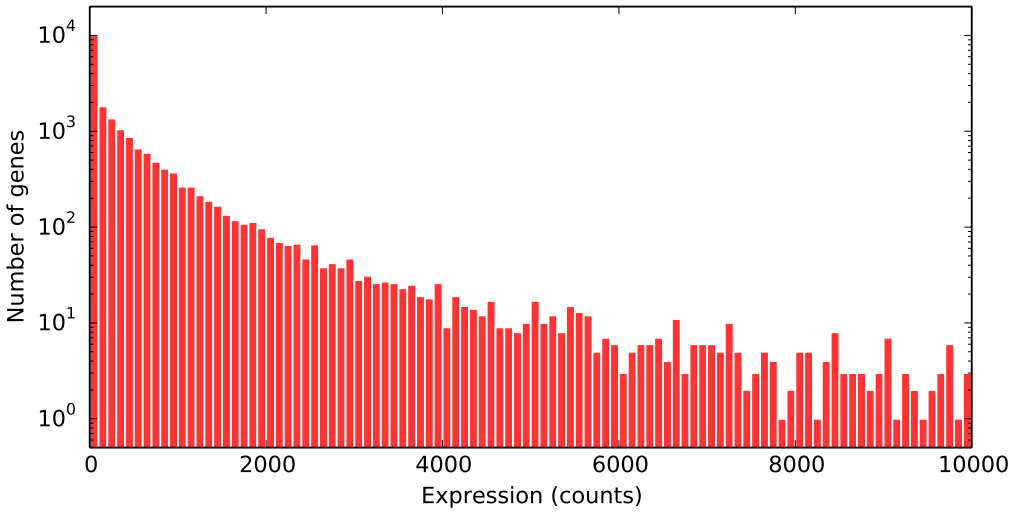


(a)

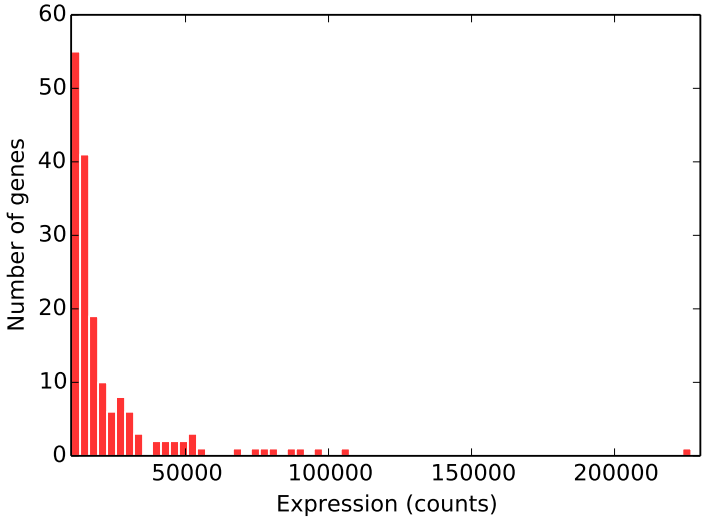


(b)

Figure 2.15: **Mean-variance relationship of estimated counts.** We use scatter plot and density plot to show the dependence between mean and variance of the inferred transcript expression levels. The transcript expression level variance is at least as high as the mean expression due to random sampling, and is further increased due to read ambiguity.

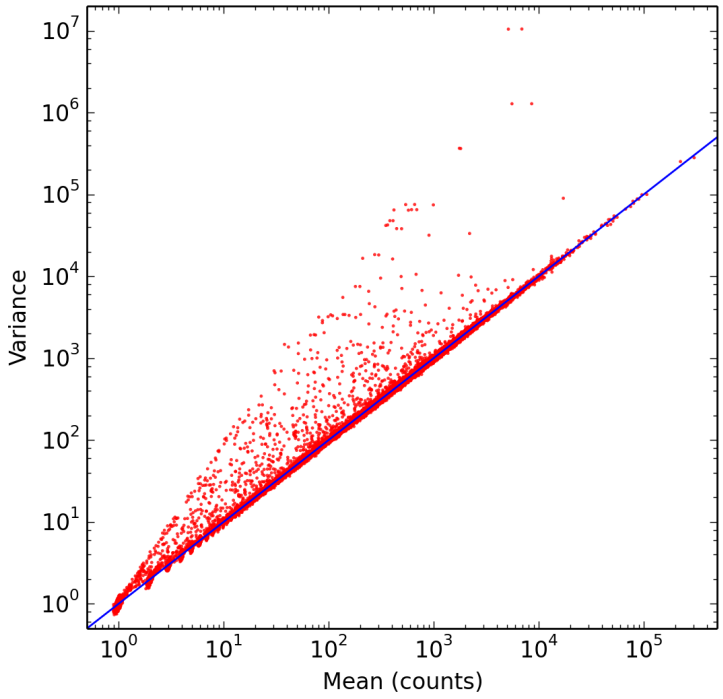


(a)

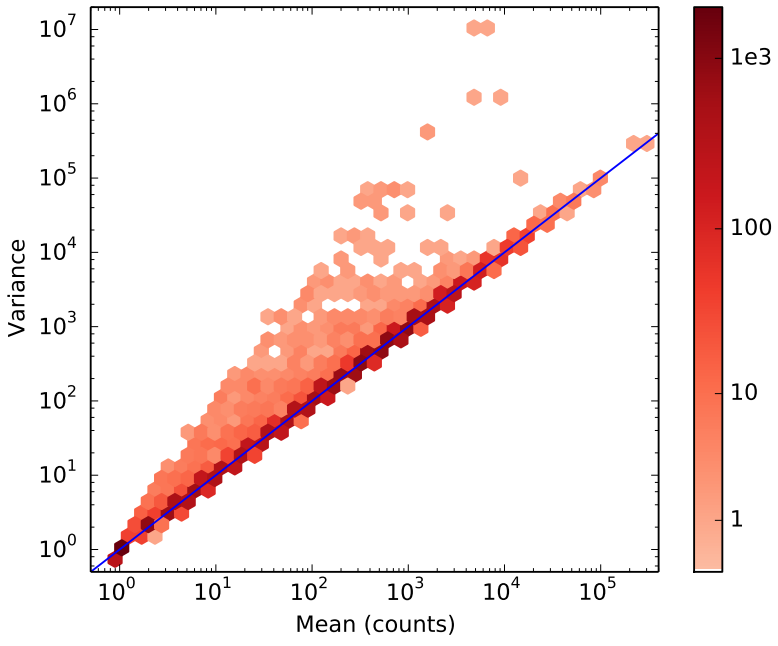


(b)

Figure 2.16: **Histograms of mean gene expression levels in counts.** The entire range is split into two histograms: (a) majority of genes have estimated read count below 10^4, we use logarithmic y-axis, (b) the top 170 expressed genes are spread over large expression range.



(a)



(b)

Figure 2.17: **Mean-variance relationship of estimated gene counts.** We use scatter plot and density plot to show the dependence between mean and variance of the converted gene expression levels. The variance of gene expression is lower than the transcript expression variance presented above.

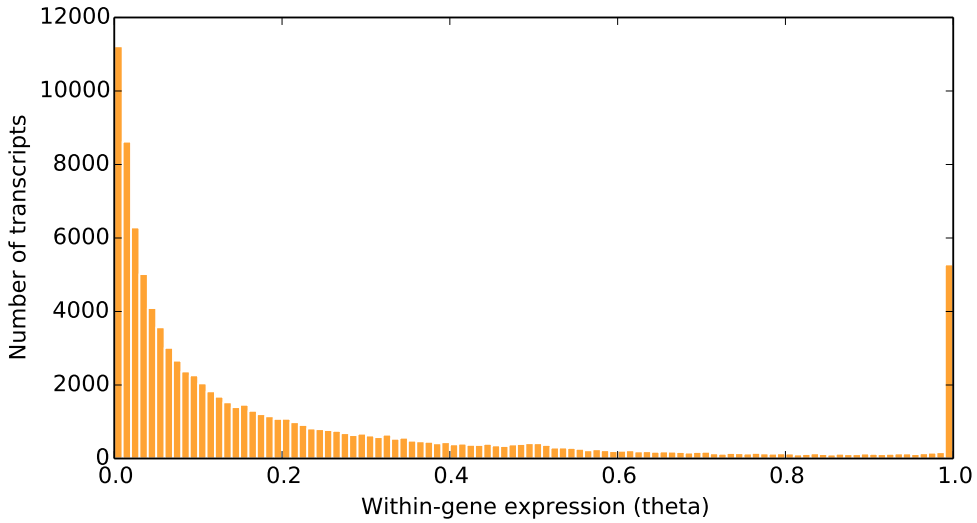


Figure 2.18: **Histogram of within-gene relative expression of transcripts.**

ambiguity which leads to correlations between transcripts. Adding expression of correlated transcripts decreases the variance of the sum.

Lastly we examined the within-gene relative expression of transcripts. Figure 2.18 shows the histogram of the proportional measure of genes' splice variants. The reference annotation contains 5119 genes with single transcript. These transcripts have relative expression 1.0 and form the majority of the peak at 1.0 within the histogram.

The mean-variance relationship of the within-gene proportions of transcripts is presented in Figure 2.19. Transcripts that are highly represented within a gene and transcripts with very low relative expression exhibit the lowest variance. In the case of mid-expressed transcripts, other splice variants of the same gene are present and 'compete' for the reads, causing increased variance.

RPKM expression

RPKM, or reads per kilobase of transcript length per million of sequenced reads, is an alternative measure that is proportional to the abundance of molecules as it is adjusted by the effective length of transcripts. It also provides a basic normalisation approach using the total number of sequenced reads.

We used Log RPKM samples of transcript (or gene) expression as an input to our differential expression analysis method. Here we show the expression levels of transcripts converted into Log RPKM. Figure 2.20 shows histogram of the mean

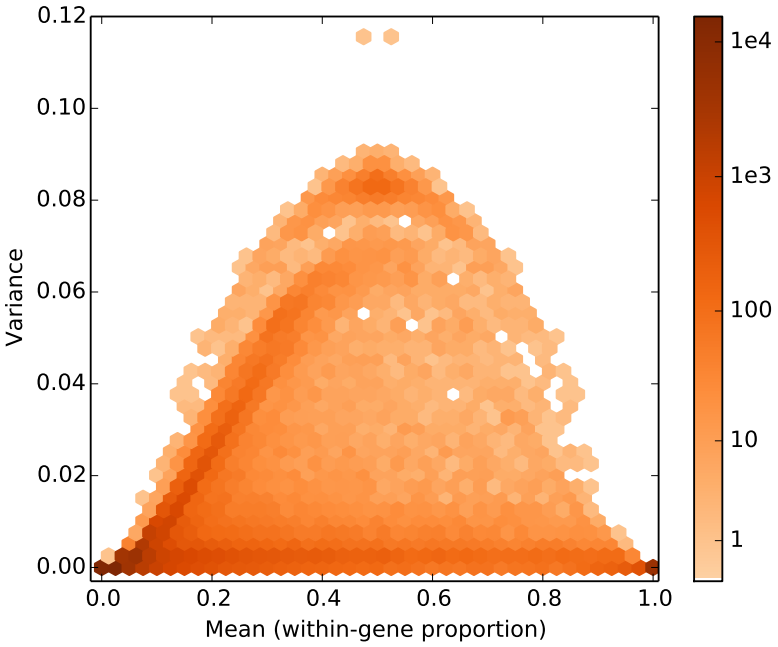


Figure 2.19: **Mean-variance relationship of within-gene relative expression of transcripts.** We use density plot to show the dependence between mean and variance of the converted within-gene relative expression levels.

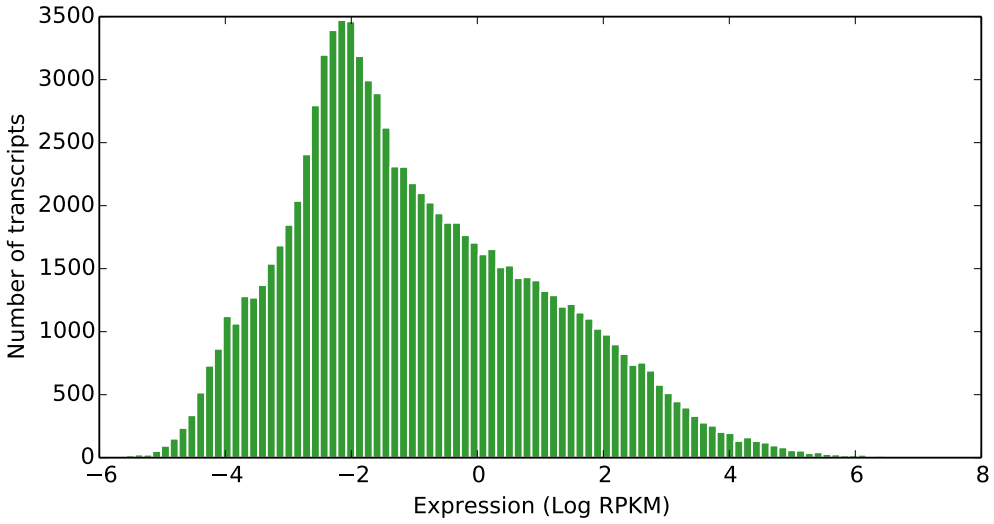


Figure 2.20: **Histogram of mean transcript expression levels in Log RPKM.**

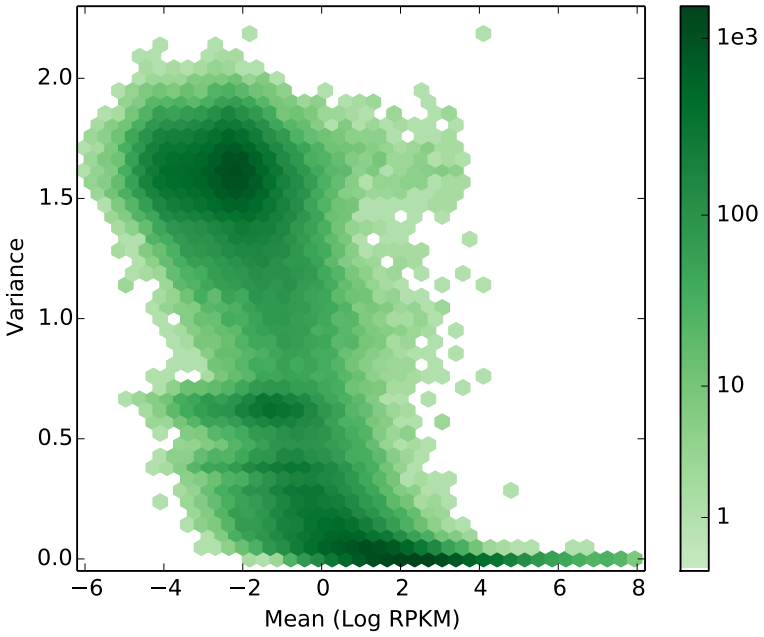


Figure 2.21: **Mean-variance relationship of estimated transcript Log RPKM.** RPKM conversion and subsequent logarithm transform the shape of the posterior distributions of expression. The logarithm transform stabilises the variance of expression levels.

Log RPKM expression.

Computing the Log RPKM involves dividing $\boldsymbol{\theta}$ by the effective length, multiplying by a constant and taking the logarithm of the result. Both division by the length and logarithm transform the variance of the resulting samples. The effect is demonstrated in Figure 2.21 showing mean-variance relationship for Log RPKM samples.

Correlations between transcripts

In Section 2.4.2 we examined correlations between transcript expression levels inferred from the Xu et al. (2010) dataset. While we used different reference sequence for the analysis, both knownGene and Gencode reference used here contain equivalent three transcripts.

The Gencode equivalent for the knownGene gene Q6ZMZ0 is denoted as ENSG00000116514.12 and similarly has three different transcripts. The transcripts are equivalent to the knownGene transcripts, with mapping denoted in

Table 2.2. The only difference is that in the Gencode annotation all three transcripts have a distinct 3' UTR.

gene	Q6ZMZ0	ENSG00000116514.12
transcript	uc010oho.1	ENST00000373456.7
	uc001bwm.3	ENST00000356990.5
	uc010ohp.1	ENST00000235150.4

Table 2.2: **Mapping of transcripts from knownGene to Gencode annotations.**

Figure 2.22 shows pairwise density plots of the three transcripts' expression samples, based on the H1-hESC data. The anti-correlation between transcripts ENST00000373456.7 and ENST00000235150.4 is not apparent, however there is distinct anti-correlation between expression levels of transcripts ENST00000356990.5 and ENST00000235150.4. Note that while in the knownGene reference transcripts uc010oho.1 and uc010ohp.1 shared the same 3' UTR, the 3' UTR of the equivalent transcripts ENST00000373456.7 and ENST00000235150.4 differs in Gencode annotation. This can explain the missing anti-correlation pattern in Figure 2.22(a).

While the previous example used the Log RPKM expression, which resulted in more pronounced anti-correlation pattern, here we use the read count which does not involve length adjustment and log transformation. Furthermore, the read count measure corresponds to the read-transcript assignments within the probabilistic model.

2.4.4 Evaluation against qRT-PCR

Exact evaluation of estimation accuracy on real RNA-seq data is difficult. There are many published RNA-seq datasets available for analysis, but the true expression for most of them is unknown. One option is to use quantitative reverts-transcription PCR (qRT-PCR) for measuring abundance of molecules of certain transcripts within sample (Roberts et al., 2011). While qRT-PCR is believed to be a more accurate measurement of the abundance, it is rarely used for more than a few hundred transcripts as it is laborious and costly.

For the evaluation of bias correction effects as well as comparison with other methods we used paired-end RNA-seq data from the Microarray Quality Control (MAQC) project (Shi et al., 2006), Short Read Archive accession number

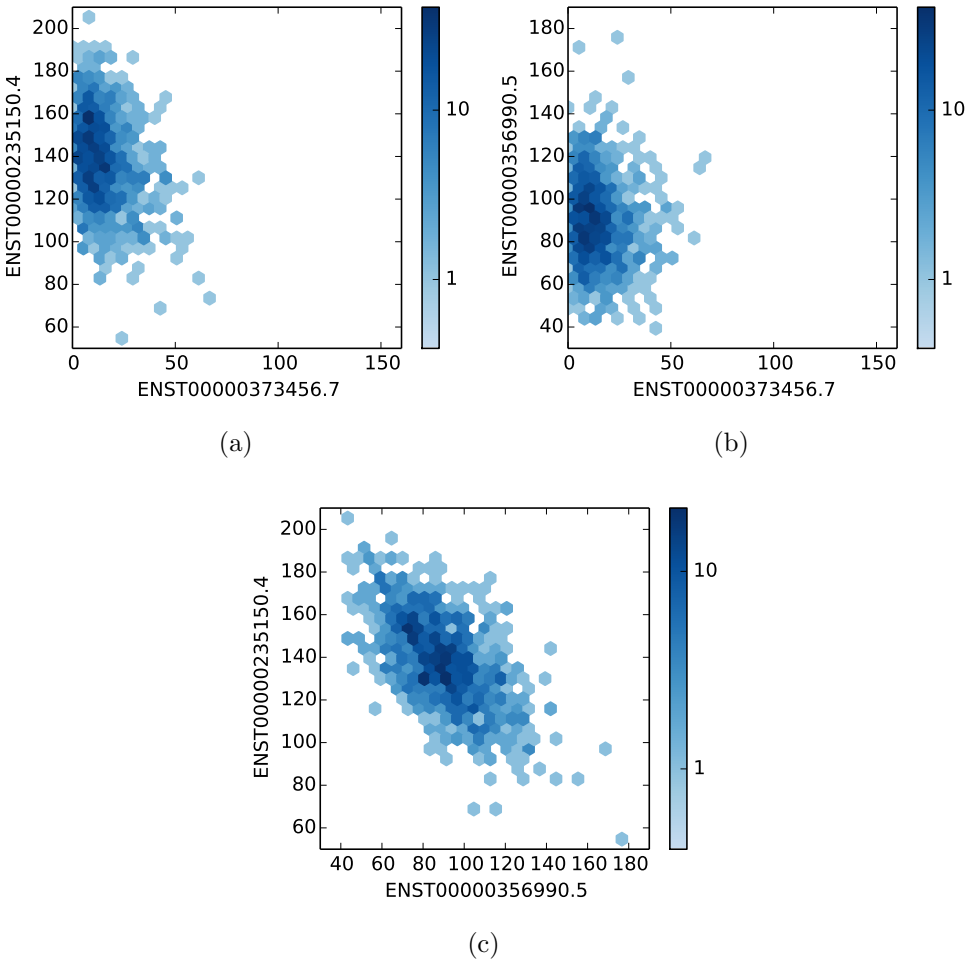


Figure 2.22: **Pairwise density plots of posterior distribution of transcript expression levels based on H1-hESC data.** Three transcripts of single gene are compared in density plots showing distribution of transcript expression measured in read counts. The transcripts are compared in the same order as equivalent transcripts in Figure 2.11.

SRA012427. 907 transcripts in this dataset were also analysed by TaqMan qRT-PCR, out of which 893 matched our reference annotation. We used RefSeq refGene transcriptome annotation, assembly NCBI36/hg18 in order to keep results consistent with the qRT-PCR data as well as previously published comparisons by Roberts et al. (2011). The reads were aligned using Bowtie (Langmead et al., 2009).

The dataset contains three technical replicates generated by three sequencing runs of the same library. The replicates were analysed separately and the resulting estimates for each method were averaged together to obtain the final expression

Bias model	Method	Average	Rep. 1	Rep. 2	Rep. 3
uniform	$*$	0.758	0.757	0.758	0.759
uniform	†	0.767	0.767	0.766	0.767
bias corrected	$*$	0.756	0.755	0.756	0.757
bias corrected	†	0.765	0.764	0.764	0.765
bias corrected	‡	**0.801**	0.801	0.795	0.804

Table 2.3: **Effects of the effective length normalisation of expression levels with respect to qRT-PCR abundance measurement.** The results are presented in terms of Pearson R^2 correlation coefficient of the 893 transcripts' expression level estimates obtained by BitSeq v0.4 and TaqMan qRT-PCR results. Three different methods of effective length normalisation were applied: $*$ – using actual transcript length, † – using effective length accounting for fragment length distribution, ‡ – using effective length accounting for fragment lengths and read distribution bias. For each method we present values for average expression taken over all three replicates as well as for each technical replicate separately.

estimates. We calculated the squared Pearson correlation coefficient (R^2) of the qRT-PCR results with the averaged expression as well as with each technical replicate.

Effective length adjustment

The qRT-PCR reports expression in terms of molecule abundance while our model represents the expression in terms of the relative expression of fragments, θ. To make the results comparable, θ has to be either converted into RPKM or normalised by transcript length. In case of BitSeq, the major improvement of accuracy originates from the use of correct effective length normalization. Using the bias corrected effective length for this conversion leads to higher correlation with the qRT-PCR. This means that using an expression measure adjusted by the effective length, such as RPKM, is more suitable for DE analysis than normalised read counts especially in cases when the read biases could differ between replicates.

We demonstrate the effects of the effective length normalisation in Table 2.3. We compared three different methods of length normalisation in combination with uniform and bias corrected expression estimates. In the first approach ($*$), the expression is adjusted just by the length of a transcript. This default method produces the poorest correlation with the qRT-PCR results.

Method	Read distribution	Average	Rep. 1	Rep. 2	Rep. 3
BitSeq	uniform	0.767	0.767	0.767	0.767
BitSeq	bias corrected	**0.798**	0.800	0.794	0.800
Cufflinks	uniform	0.758	0.758	0.759	0.757
Cufflinks	bias corrected	**0.786**	0.785	0.785	0.785
RSEM	uniform	0.763	0.762	0.762	0.764
RSEM	bias corrected	**0.763**	0.762	0.762	0.764
MMSEQ	uniform	0.761	0.760	0.760	0.762
MMSEQ	bias corrected	**0.799**	0.799	0.793	0.802

Table 2.4: **Comparison of expression estimation accuracy against TaqMan qRT-PCR data and the effect of non-uniform read distribution models.** The correlation of expression estimates with 893 matching transcripts analysed by qRT-PCR is reported using Pearson's R^2 coefficient. Correlation for each replicate, as well as the averaged expression levels were calculated for all methods using both uniform and non-uniform read distribution model.

The second approach (†) uses effective length computed based on the fragment length distribution observed from paired-end reads assuming uniform reads distribution, as defined in Equation 2.23.

The best result for this dataset yields the last approach (‡), which uses effective length dependent on the fragment length distribution as well as read distribution bias weights (see Equation 2.24). As expected, bias corrected estimates of θ normalised by lengths that do not account for non-uniform read distribution show poorer accuracy than the uniform estimates (rows 3 and 4 vs. 1 and 2). On the other hand, bias corrected expression estimates adjusted by the bias corrected effective length result in the best correlation with the qRT-PCR expression measurements.

Comparison with alternative quantification methods

The effects of correcting for read distribution biases are presented in Table 2.4 together with a comparison of three alternative transcript expression estimation methods: Cufflinks 2.1.1 (Trapnell et al., 2013), MMSEQ v0.9.18 (Turro et al., 2011) and RSEM v1.1.14 (Li and Dewey, 2011). MMSEQ and RSEM similarly use Bowtie to align the reads to the reference transcriptome, in case of Cufflinks the reads are aligned to genomic reference using splice-aware alignment tool TopHat (Trapnell et al., 2009).

Our results show that using uniform read distribution model yields similar

level of accuracy for all four methods, with BitSeq performing slightly better than the other three methods. Correcting for read distribution bias can further improve the expression estimates.

Here BitSeq, Cufflinks and MMSEQ use the same method for read distribution bias correction and provide improvement over the uniform model similar to improvements previously reported by Roberts et al. (2011). We were not able to use the default bias correction provided by MMSEQ due to an error in an external `R` package `mseq` used for estimation of transcript effective lengths. Instead, we provided the MMSEQ package with effective lengths computed by BitSeq bias correction algorithm in order to produce the results for this comparison. The bias correction of mseq package itself was already compared against Cufflinks on the same dataset showing slightly worse accuracy and less improvement (Roberts et al., 2011).

The RSEM package uses its own method for bias correction based on the relative position of fragments, which in this case did not improve the expression estimation accuracy for the selected transcripts.

2.4.5 Evaluation on synthetic data with uniform read distribution

For further evaluation of the accuracy of our estimation method we used synthetic data generated based on observed properties of previously analysed RNA-seq datasets. The synthetic data provides the great advantage of known ground truth expression and can be easily generated through simulation of the sequencing process. It also captures the main difficulty of transcript expression quantification, that is the ambiguity of transcripts. As reads are sampled randomly along entire reference transcripts, many reads will align back to multiple transcripts as in a real RNA-seq experiment.

The downside of simulated data is that it is difficult to replicate the read distribution biases. The biases towards sequencing specific fragments can be caused by various factors of RNA sample preparation and NGS technology and their causes are still not fully documented. In this section we use simulated data that was generated by uniform sampling of reads along transcripts without any kind of bias.

Synthetic data generation

The reads were generated from the UCSC NCBI37/hg19 knownGene transcript annotation (Hsu et al., 2006; Meyer et al., 2013), consisting of 27297 genes with 77614 transcripts in total. The initial expression levels were based on expression estimates from the Xu et al. (2010) dataset. The dataset was further used to estimate the fragment size distribution for paired-end reads $l_f \sim \text{LogNorm}(5.32, 0.12)$ and to estimate the empirical distribution of Phred quality scores of individual bases. A sequence of quality scores was generated for each read and base substitutions were added with the probability given by the Phred score of each base.

Fragments were assigned to transcripts according to a multinomial distribution parametrised with the ground truth relative expression of transcript fragments (θ). Fragment positions within transcripts were sampled uniformly from all possible position given a fragment length. 10 million fragments were sampled in total, with reads generated in pairs from both ends of each fragment.

Estimation accuracy

The synthetic reads were aligned to the UCSC knownGene transcriptome reference using Bowtie (Langmead et al., 2009) and transcript expression levels were estimated using BitSeq. We again include comparison with three other methods for transcript expression quantification: Cufflinks 2.1.1 (Trapnell et al., 2013), MMSEQ v0.9.18 (Turro et al., 2011) and RSEM v1.1.14 (Li and Dewey, 2011). All methods were used without the read distribution bias correction, as the data was generated through uniform sampling of fragment positions within transcripts.

We evaluated the accuracy of the four methods in terms of correlation with the known ground truth in three different expression measures. Firstly, we compared the transcript RPKM as an absolute transcript expression measure. Secondly, we used relative within-gene expression of transcripts which expresses the relative proportion of a transcript within transcripts of the same gene. Finally we compared the gene RPKM, calculated by adding up the transcript expression levels for each gene.

The results in form of Pearson's R^2 correlation coefficient are presented in Table 2.5. We can see that our model, MMSEQ and RSEM provide very high correlation with the known ground truth for both transcript and gene absolute expression. For the relative within-gene expression levels, BitSeq is more accurate than the other methods. In spite of providing slightly better results in the absolute

	Measure (cutoff)			
Method	transcript (1)	relative (10)	relative (100)	gene (1)
BitSeq	0.994	**0.945**	**0.963**	0.994
Cufflinks	0.826	0.829	0.897	0.838
RSEM	0.995	0.876	0.946	0.996
MMSEQ	**0.997**	0.886	0.948	**0.998**

Table 2.5: **The R^2 correlation coefficient of estimated expression levels and the ground truth.** Three different expression measures were used: absolute transcript expression, relative within-gene transcript expression and gene expression. Comparison includes only sites with at least 1 read per transcript for transcript expression, either 10 or 100 reads pre gene for within-gene transcript expression and at least 1 read per gene for gene expression.

measure, RSEM and MMSEQ show worse correlation in the relative within-gene measure as they tend to assign zero expression to some transcripts within one gene. This is most likely caused by the use of Maximum Likelihood parameter estimates as the starting point for the Gibbs sampling algorithm.

We further present density scatter plots of expression estimates plotted against the known ground truth in Figure 2.23. BitSeq overestimates expression for a fraction of transcripts, however, this is largely the effect of the RPKM transformation and the use of uniform prior $\alpha^{dir} = 1$. The prior distribution assigns a pseudo-count of one read per transcript, leading to non-zero expression for all transcripts. The subsequent length normalisation in RPKM measure results in increased RPKM of weakly-expressed short transcripts. Nevertheless, the high variance of expression levels for such transcripts signifies the uncertainty of such estimate and prevents from false positive differential expression calls.

In Figure 2.23 we can further observe high level of underestimation of transcript expression levels in Cufflinks, RSEM and MMSEQ results which are avoided when using BitSeq.

2.4.6 Comparison of sampling algorithms

We implemented two variations of Gibbs sampling algorithm, standard Gibbs sampling and collapsed Gibbs sampling. The standard Gibbs sampling algorithm samples the read assignment $\boldsymbol{I}$, relative expression $\boldsymbol{\theta}$ and noise parameter θ^{act} from conditional distributions in each iteration. The collapsed Gibbs sampling

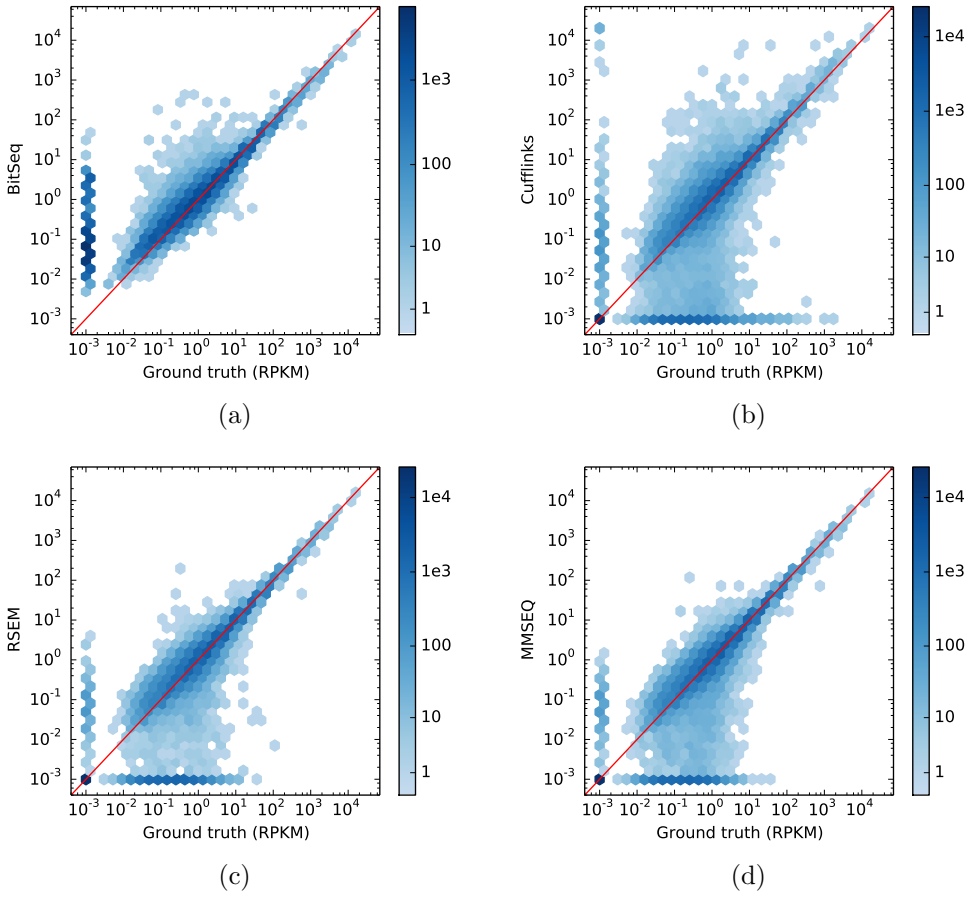


Figure 2.23: **Comparison of expression estimates using 10M simulated paired-end reads with known expression.** Our model implemented in BitSeq (a), Cufflinks (b), RSEM (c) and MMSEQ (d) are compared in terms of the absolute transcript expression. The expression estimates were converted into RPKM for each transcript and compared against the ground truth using density scatter plots with logarithmic scale. RPKM below 10^{-3} was changed to 10^{-3} in order to fit within the plot.

algorithm, is derived from 'collapsed' model. In the collapsed model, the relative expression $\boldsymbol{\theta}$ and noise parameter θ^{act} are marginalised, leaving just the read assignments. In each iteration, individual read assignments are sampled from conditional distributions conditioned on all other assignment. The relative expression $\boldsymbol{\theta}$ is only sampled when output is being generated. The marginalization of $\boldsymbol{\theta}$ and θ^{act} leads to faster convergence of the algorithm.

We investigate the convergence advantage of the collapsed sampler on the RNA-seq data from H1-hESC cell line generated by the ENCODE consortium (Djebali et al., 2012). Briefly, the data contains 14.5M mapped read pairs with

62.3M alignments. The read alignment likelihoods were estimated using the non-uniform read distribution model.

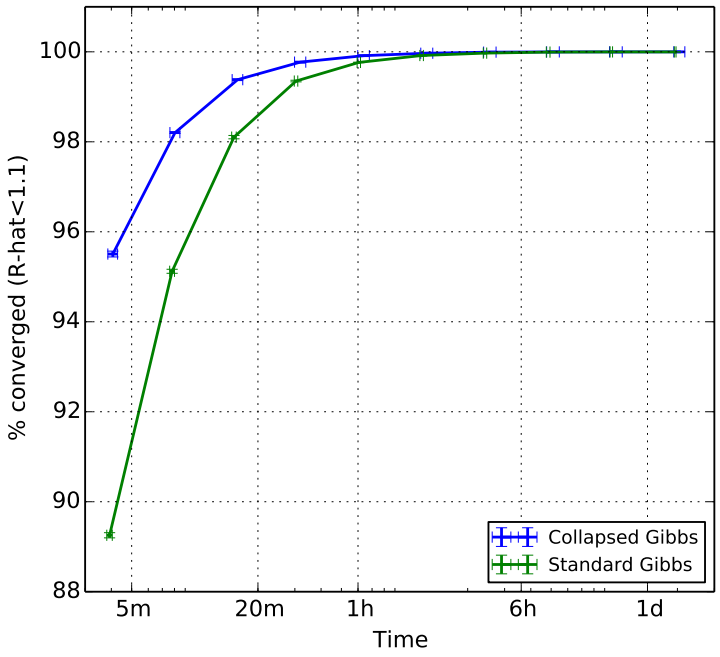
We ran both standard Gibbs sampler and collapsed Gibbs sampler using six parallel Markov chains. Starting with 100 samples as burn-in and generating 100 samples for convergence checking, the number of generated samples is doubled in following iterations up-to top limit of 50K samples. Convergence is evaluated by the $\widehat{R}$ statistic of marginal distributions of transcript expression levels, as described in Section 2.3.4.

The convergence of both methods dependent on time is shown in Figure 2.24. While the majority of transcripts converge within very few iterations (Figure 2.24(a)), we want as many transcripts to converge as possible. In the second panel we show the mean of the ten highest $\widehat{R}$ values over all transcripts. We can observe from both figures that while the standard Gibbs sampler has slightly faster run time in terms of constant number of iterations, it needs more than twice as many iterations to reach comparable level of convergence. The collapsed Gibbs sampler converges much faster than the standard method and thus decreases the number of necessary iterations.

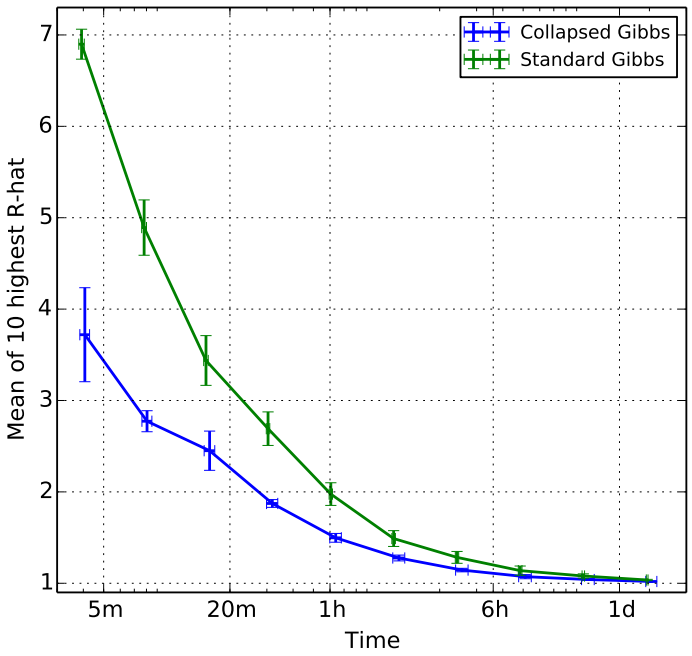
2.4.7 Evaluation of computational requirements

Our inference procedure uses the Gibbs sampling algorithm for generating samples from the posterior distribution. Such approach can lead to computationally expensive calculations due to slow convergence and time used for generation of every sample. Thanks to efficient implementation and parallelisation, our application can process 100 million paired-end reads in just under 30 hours using 4 CPUs and 8.5GB of memory. Given that the data acquisition for a single experiment lasts multiple days, the computational cost necessary for the analysis can be neglected.

Further evaluation of time and memory requirements of the inference algorithm are presented in sections 4.2.1, 4.2.3 and 4.2.4.



(a)



(b)

Figure 2.24: **Convergence evaluation of Gibbs sampler and collapsed Gibbs sampler.** The convergence is evaluated using the $\widehat{R}$ statistic, results being averaged over five runs using the standard deviation for errorbars. (a) the percentage of converged transcripts, based on $\widehat{R} < 1.1$, (b) mean $\widehat{R}$ of 10 least converged transcripts.

2.5 Summary

We have developed a novel application for transcript expression level quantification from RNA-seq data. The method is based on a generative probabilistic model of the data and uses Bayesian methodology to infer the expression given a set of observed reads. The probabilistic model considers read errors, fragment lengths and read distribution biases when calculating probabilities for individual alignments. We use the Gibbs sampling algorithm to generate samples from the posterior distribution of transcript expression levels.

The major advantage of our method is that it infers the full posterior distribution of expression levels. The posterior can be summarized by its mean value, while variance and percentiles can be used to assess the certainty of our estimate. We can further look at the anti-correlations between individual expression levels, which reflect the similarity of respective transcripts. Lastly, in the next chapter we introduce a novel differential expression analysis approach which utilises the full posterior distribution of transcript expression levels.

The evaluations on real and simulated RNA-seq data showed high accuracy of the expression estimates. We compared our approach with alternative methods for transcript abundance quantification showing comparable accuracy with other state-of-the-art applications, while providing a reliable approach for read distribution bias correction.

Chapter 3

Detecting changes in transcript expression

In this chapter we present a new approach for differential expression analysis. The aim of the DE analysis is to compare gene and transcript abundances between two or more conditions in order to identify those that exhibit significant changes of abundance between conditions.

We propose a probabilistic model of estimated transcript expression from different conditions. The model relies on data from biological replicates within a condition to estimate the natural fluctuations of abundances due to differences in the underlying biological and experimental conditions.

The novelty of this approach is that it uses inferred posterior distributions of transcript expression instead of just a single point estimate. Hence the confidence of the expression estimate can be included when determining significance of expression changes. The method uses samples from the posterior distribution generated by the expression quantification algorithm presented in Chapter 2.

In our analysis, we are primarily interested in changes of expression of transcripts, e.g. specific gene isoforms. However, our method can be equally applied to gene-level DE analysis, using combined expression of associated transcripts.

Probability of positive log ratio When comparing expression of two conditions we are provided with MCMC samples from the posterior distribution of expression of each transcript. As the posterior does not have an analytic form we will use the MCMC samples for the comparison in order to utilize the full posterior. We will use the expression $\vartheta_{m,n}^{(c_1)}$ to denote the n-th MCMC expression

samples of the m-th transcript of condition c_1. For now $\boldsymbol{\vartheta}$ will generally denote expression level, without specifying the measure we are using, which could be either proportion of transcript fragments $\boldsymbol{\theta}$, $RPKM$ or some other representation of transcript expression.

We use a one-sided Bayesian test for DE that has been previously used in microarray DE analysis (Hein and Richardson, 2006; Liu et al., 2006). Liu et al. (2006) refer to it as the Probability of Positive Log-Ratio (PPLR) statistic, which we adopted in this work as well. For a transcript m, the PPLR is defined as

$$PPLR_m = P\left(\log \frac{\vartheta_m^{(c1)}}{\vartheta_m^{(c2)}} > 0 \middle| R\right) = P\left(\vartheta_m^{(c1)} > \vartheta_m^{(c2)} | R\right) \tag{3.1}$$

and provides a simple way of expressing the probability of expression in one sample being higher than the expression in the other sample, so called up-regulation. Furthermore, PPLR can be easily estimated by comparing two sets of MCMC samples by following expression:

$$\mathrm{E}[PPLR_m] = \frac{1}{N}\sum_{n=1}^{N} \delta\left(\vartheta_{m,n}^{(c1)} > \vartheta_{m,n}^{(c2)}\right), \tag{3.2}$$

we use $PPLR_m$ for shorthand notation.

The main advantage of using PPLR for determining significance of DE is that it assesses the entire distributions of the estimated expression. Consider an example in Figure 3.1. While the average log fold change, $\widehat{\log FC}$, of both examples is the same, PPLR in the first example (3.1(a)) is lower than PPLR in the second example (3.1(b)), where it is close to 1. The lower DE significance of the first example is what we would expect as the uncertainty of expression estimates is increased.

Biological variance In order to produce reliable estimates of differential expression we have to account for biological variance, the fluctuation of expression levels of transcripts and genes within the same condition (Auer and Doerge, 2010). These occur naturally as the abundance of transcript molecules in cells and tissues varies and were reported in microarray DE analysis as well (Dudoit et al., 2002).

The biological variance is transcript and condition dependent and thus cannot

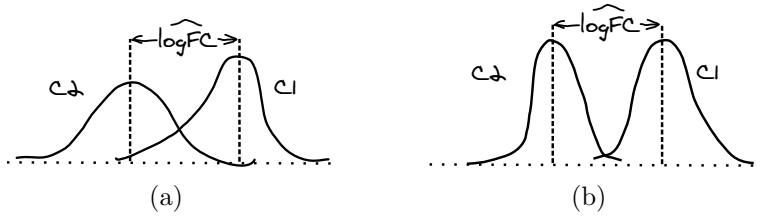


Figure 3.1: **Illustrative example of using PPLR vs log fold change for DE testing.** In both cases, the difference between mean expression of the two conditions is the same, leading to the same average log fold change. PPLR is different for each example: (a) PPLR is decreased due to the overlap of the two distributions, (b) PPLR is close to 1 as the two distributions have minimal overlap.

be accounted for *a priori*. For this reason it is essential to perform DE analysis using biological replicates. Unlike technical replicates, which involve multiple sequencings of the same sample or even the same library, biological replication means analysis of multiple independent samples from the same condition. With biological replicates it is possible to assess the amount of variation within each condition and account for it when selecting functional changes of transcript expression.

3.1 Probabilistic model of Differential Expression

We propose a probabilistic model of expression gathered from replicates of each condition in order to infer per-condition mean expression levels. This model enables us to assess the biological variance observed from replicates of each condition and account for it before calling differentially expressed transcripts. We use a hierarchical Log-Normal model of transcript expression levels observed in replicates of multiple conditions. To simplify the derivation, we consider a logarithmic transformation of expression levels, denoting $y_m = \log \vartheta_m$, and use a Normal model over y_m.

As the expression levels in RPKM or other measure are always positive, we apply the log transformation. This enables the use of the Normal distribution to model the expression dependence and further allows the use of conjugate probabilistic model. The logarithmic transformation also stabilises the variance of

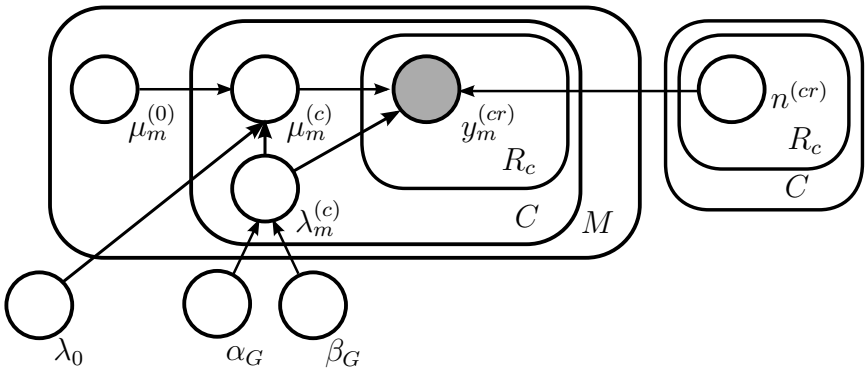


Figure 3.2: **Graphical model of transcript expression estimates with biological variance from multiple conditions.** For replicate r, condition c and transcript m, the observed log-expression level $y_m^{(cr)}$ is normally distributed around the normalised condition mean expression $\mu_m^{(c)} + n^{(cr)}$ with biological variance $1/\lambda_m^{(c)}$. The condition mean expression $\mu_m^{(c)}$ for each condition is normally distributed with overall mean expression $\mu_m^{(0)}$ and scaled variance $1/(\lambda_m^{(c)}\lambda_0)$. The inverse variance, or precision $\lambda_m^{(c)}$, for a given transcript m follows a Gamma distribution with expression-dependent hyperparameters α_G, β_G, which are constant for a group of transcripts G with similar expression.

expression levels, which leads to more stable estimates of the condition specific expression.

Graphical representation of the model is depicted in Figure 3.2. The model assumes expression samples from C conditions with R_c replicates per condition. The log-expression from replicate r, $y_m^{(cr)}$ is assumed to be distributed according to a normal distribution with condition mean expression $\mu_m^{(c)}$ and condition specific precision $\lambda_m^{(c)}$. The expression of every transcript in each replicate is normalised by a replication specific constant $n^{(cr)}$ which allows for additional adjustment of varying sequencing library size. The normalisation constant can be estimated prior to probabilistic modeling using, for example, a quantile based method of Robinson and Oshlack (2010) or any other suitable technique.

The condition mean expression of each condition is normally distributed with mean $\mu_m^{(0)}$ and scaled precision $\lambda_m^{(c)}\lambda_0$. The overall mean transcript expression $\mu_m^{(0)}$ is empirically estimated before application of the model. Instead of using separate parameter for the precision of condition mean expression, we chose to use the scaled within-condition precision. This simplifies the model in terms of number of parameters and relates the biological variance and variation between conditions. With increased biological variance within one condition we expect

to observe higher variance between conditions. The scaling factor is kept as a parameter of the inference procedure with default value set to $\lambda_0 = 2$.

The prior distribution over per-transcript, condition-specific precision $\lambda_m^{(c)}$ is a Gamma distribution with hyperparameters α_G, β_G, which are fixed for a group of transcripts with similar expression level, G. We detail the hyperparameter inference in Section 3.2.1.

We summarize the model definition in the following expressions outlining the probability distributions over the parameters:

$$y_m^{(cr)} \sim \text{Norm}(\mu_m^{(c)} + n^{(cr)}, 1/\lambda_m^{(c)}), \tag{3.3}$$

$$\mu_m^{(c)} \sim \text{Norm}(\mu_m^{(0)}, 1/(\lambda_0 \lambda_m^{(c)})), \tag{3.4}$$

$$\lambda_m^{(c)} \sim \text{Gamma}(\alpha_G, \beta_G). \tag{3.5}$$

As it is impossible to relate various transcripts in terms of their differential expression, we apply the model on per-transcript basis. On the other hand, we have observed that the significance of biological variance is expression dependent with similar effects over all transcripts. To use the shared information about biological variance, we use expression dependent hyperparameters for the prior distribution over precision. The model considers each transcript independently and does not put any relation on transcripts from one gene or other closely related genes, but uses hyperparameters α_G and β_G, that are inferred for a specific group of transcripts of similar expression.

The independence assumption does not affect transcripts' individual differential expression calls. However, as the expression levels of certain transcript can be correlated, it can lead to underestimation of overall false discovery rate as we discuss later.

The DE model is conjugate and therefore the inference can be carried out exactly, making the whole process computationally tractable.

Technical noise propagation The marginal distribution of expression for one replicate cannot be always well approximated by a Log-Normal distribution. This is due to intrinsic technical noise, originating from discrete assignments of reads to transcripts in the probabilistic model and the sequencing technology. For an illustration, please see Figures 2.10 or 3.3(a) depicting the marginal posterior distributions of transcript expression.

We propagate the technical noise without the restraint of parametric probability distribution by the novel approach of applying the DE model to individual MCMC samples. The posterior samples obtained by MCMC sampling during the transcript expression analysis can be considered as *pseudo-data*. We construct a pseudo-data vector using a single MCMC sample for each replicate across all conditions. The posterior distribution over per-condition means is inferred for each pseudo-data vector using the model in Figure 3.2. We then use Bayesian model-averaging to combine the evidence from each pseudo-data vector and determine the probability of differential expression. This effectively regularizes our variance estimate in the case that the number of replicates is low. As shown in Section 3.4.1 this provides improved control of error rates for weakly expressed transcripts where the technical variance is large.

3.2 Model inference

The model depends on the expression level dependent hyperparmeters α_G and β_G and mean expression $\mu_m^{(0)}$. The $\mu_m^{(0)}$ represents a broad estimate of average expression of transcript m. We calculate it by taking mean over all replicates from every condition using log-transformed MCMC samples of transcript expression,

$$\mu_m^{(0)} = \sum_{c=1}^{C} \sum_{r=1}^{R_c} \sum_{n=1}^{N} \log \vartheta_m^{(cr)(n)} + n^{(cr)}. \tag{3.6}$$

We use the overall mean expression to divide transcripts into groups with similar expression. The number of groups is parameter of the inference procedure with default setting of 200 groups. Both technical and biological variance change with expression. While technical variance decreases for transcripts with higher expression level, the differences over biological replicates are more pronounced. We demonstrated this effect already in Chapter 2 in Figure 2.13, which compares the averaged variance of Log RPKM expression of single sample, technical replicates and biological replicates. The significance of biological variance increases for highly expressed transcripts.

The transcripts are grouped based on $\mu_m^{(0)}$ and we infer the hyperparameters α_G, β_G jointly for the whole group. Once the hyperparameters are estimated, the inference of condition mean expression can be carried out independently for each transcript.

3.2.1 Inference of expression dependent priors

We build upon the observation of biological variance being expression dependent. The variance of expression levels of higher-expressed transcripts are increased above the technical variance of the sampling process. To capture this dependence, we use expression-dependent hyperparameters over precision of the Normal distribution. The hyperparameters are inferred from the model (Figure 3.2) by marginalizing over condition mean expression and condition specific precision.

Firstly, the overall expression given by $\mu_m^{(0)}$ of all transcripts is divided into bins of equal expression range. Transcripts are assigned to these bins based on their expression $\mu_m^{(0)}$ forming groups of transcripts with similar expression. The inference of hyperparameters for group G is carried out jointly for all M_G transcripts using pseudo-data vectors consisting of individual MCMC samples from all replicates.

We use vague uninformative prior over the hyperparameters in the form of a uniform distribution.

The posterior distribution over the hyperparameters does not have a tractable analytic form. In the derivation, we omit the normalisation constant and only express the posterior in terms of proportionality:

$$
\begin{aligned}
P(\alpha,\beta|\boldsymbol{y}) &\propto P(\alpha,\beta)P(\boldsymbol{y}|\alpha,\beta) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} P(\boldsymbol{y}_m^c|\alpha,\beta) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} \int \mathrm{d}\lambda_m^{(c)}\, P(\lambda_m^{(c)}|\alpha,\beta) \int \mathrm{d}\mu_m^{(c)}\, P(\mu_m^{(c)}|\lambda_m^{(c)}) \prod_{r=1}^{R_c} P(y_m^{(cr)}|\lambda_m^{(c)},\mu_m^{(c)}) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha+R_c)}{\left(\beta+\frac{1}{2}\left(\lambda_0{\mu_m^{(0)}}^2+y_m^{2(c+)}-\frac{(\lambda_0\mu_m^{(0)}+y_m^{(c+)})^2}{\lambda_0+R_c}\right)\right)^{\alpha+R_c}}, \qquad (3.7) \\
y_m^{(c+)} &= \sum_{r=1}^{R_c} y_m^{(cr)}, \\
y_m^{2(c+)} &= \sum_{r=1}^{R_c} {y_m^{(cr)}}^2.
\end{aligned}
$$

Applying the Random walk Markov chain Monte Carlo

Due to the intractable normalisation constant of the posterior distribution of hyperparameters, we have to resort to the use of approximate inference algorithms. We use Markov chain Monte Carlo (MCMC) to produce samples from the posterior of each group of transcripts. In this case, we are not able to use conditional distributions of individual parameters and thus cannot use the Gibbs sampler algorithm. We apply the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970), which was introduced in Section 1.5.5.

We implement the approach of Random walk MCMC, using a Normal distribution with reflective bound at 0 as a proposal for both α_G and β_G, which can only be positive. Following from Equation 3.7, the acceptance of newly proposed α_G^* and β_G^* given previous samples α_G^{t-1} and β_G^{t-1} is

$$p_{accept} = \min\left(1, \prod_{m=1}^{M}\prod_{c=1}^{C} \frac{(\beta_G^*)^{\alpha_G^*}}{(\beta_G^{t-1})^{\alpha_G^{t-1}}} \frac{\Gamma(\alpha_G^{t-1})}{\Gamma(\alpha_G^*)} \frac{\Gamma(\alpha_G^* + R_c)}{\Gamma(\alpha_G^{t-1} + R_c)} \frac{\left(\beta_G^{t-1} + \mathcal{Y}_{mc}\right)^{\alpha_G^{t-1}+R_c}}{(\beta_G^* + \mathcal{Y}_{mc})^{\alpha_G^*+R_c}}\right), \tag{3.8}$$

$$\mathcal{Y}_{mc} = \frac{1}{2}\left(\lambda_0 {\mu_m^{(0)}}^2 + y_m^{2(c+)} - \frac{(\lambda_0 \mu_m^{(0)} + y_m^{(c+)})^2}{\lambda_0 + R_c}\right).$$

To assess the quality of a proposal distribution, it is important to monitor the acceptance rate of proposed samples. High acceptance rate signifies a conservative proposal that does not explore the full parameter space (Gelman et al., 2004). On the other hand, very low acceptance rate results in stationary Markov chain as the newly proposed samples are rejected too often. For Random walk MCMC with Normal distribution as the proposal distribution, 0.44 is an optimal acceptance rate for single variable sampling, while an acceptance rate close to 0.234 is optimal for high-dimensional multivariate case (Roberts et al., 1997; Sherlock and Roberts, 2009).

We proceed with the inference as follows, the variables ${\alpha_G}^0$, ${\beta_G}^0$ are initialized from a uniform distribution over broad interval starting at 0. After burn-in phase of the algorithm, we generate fixed number of samples from the Markov chain and inspect the acceptance rate. If the acceptance rate is within expected range, the desired number of samples is generated using the current proposal distribution. In case of very high acceptance rate, the variance of the proposal distribution is increased. On the other hand, in case of low acceptance rate, the variance of the

proposal distribution is decreased. After adjusting the proposal distribution we generate a new set of samples, while observing the acceptance rate. We repeat this procedure until reaching a proposal distribution with a desirable acceptance rate.

Smoothing of hyperparameter distribution

The MCMC algorithm described above produces samples of α_G and β_G for each group of transcripts G within the same expression range. These distributions vary even for groups of transcripts with similar expression. As a consequence, two transcripts with small difference in expression that were put into two distinct groups will have distinct distributions of hyperparameters. In order to overcome this discontinuity and also to avoid sampling from an empirical distribution of hyperparameters, we use smoothing.

We apply a non-parametric smoothing procedure Lowess (Cleveland, 1979). Each hyperparameter is smoothed separately using all samples generated by the MCMC algorithm. The smoothing is done with respect to expression, where each sample from a given group of transcripts is assigned the average expression of that group. Using this method, we obtain smoothed sequences of α_G and β_G with one pair of hyperparameters per group of transcripts. While alternative smoothing algorithms could have been applied, the Lowess algorithm is sufficient for ensuring that transcripts with similar mean expression level are used with similar hyperparameters.

The expression-dependent hyperparameters are used to share common behavior of variance between various transcripts. Adjustments to the Lowess smoothing and transcript grouping produce slight variations in the smoothed hyperparameter sequence. However, as the relation between expression and significance of biological variance is consistent, these changes have limited effect on the final differential expression calls.

3.2.2 Per sample estimation from Normal-Gamma model

The inference of the model is straightforward due to the conjugacy of the Normal model with Gamma distributed prior over precision. The model is applied to the pseudo-data vectors of MCMC expression samples using one sample per replicate, $\boldsymbol{y_m} = (y_m^{(1,1)}, y_m^{(1,2)}, \ldots, y_m^{(C,R_C)})$. We use Bayes' theorem to derive the posterior

over vectors of condition mean expression and condition specific variance for a particular pseudo-data vector:

$$P(\boldsymbol{\mu}_m, \boldsymbol{\lambda}_m | \boldsymbol{y}_m) \propto P(\boldsymbol{y}_m | \boldsymbol{\mu}_m, \boldsymbol{\lambda}_m) P(\boldsymbol{\mu}_m) P(\boldsymbol{\lambda}_m), \tag{3.9}$$

$$P(\boldsymbol{\mu}_m, \boldsymbol{\lambda}_m | \boldsymbol{y}_m) \sim \prod_{c=1}^{C} \text{Gamma}\left(\lambda_m^{(c)} \,\middle|\, \mathrm{a}_{m,c}, \mathrm{b}_{m,c}\right) \text{Norm}\left(\mu_m^{(c)} \,\middle|\, \mathrm{m}_{m,c}, \mathrm{p}_{m,c}^{-1}\right), \tag{3.10}$$

$$\begin{aligned}
\mathrm{a}_{m,c} &= \alpha_G + \frac{R_c}{2}, \\
\mathrm{b}_{m,c} &= \beta_G + \frac{1}{2}\left(\lambda_0 {\mu_m^{(0)}}^2 + y_m^{2(c+)} - \frac{(\lambda_0 \mu_m^{(0)} + y_m^{(c+)})^2}{\lambda_0 + R_c}\right), \\
\mathrm{m}_{m,c} &= \frac{\lambda_0 \mu_m^{(0)} + y_m^{(c+)}}{\lambda_0 + R_c}, \\
\mathrm{p}_{m,c} &= \lambda_m^{(c)}(\lambda_0 + R_c).
\end{aligned}$$

For detailed derivation please refer to Section A.2. The factorization of the posterior distribution enables sampling of the precision and mean sequentially. All condition specific precisions $\boldsymbol{\lambda}_m$ are sampled from C independent Gamma distributions with parameters a_c and b_c. The precision parameters are then used to sample condition specific means from C independent Normal distribution parametrised as above.

The inference for transcript m is carried out as follows. Given its overall mean expression $\mu_m^{(0)}$, we select precision hyperparameters α_G and β_G from the smoothed sequence obtained during the hyperparameter estimation step described above. The MCMC expression samples from all replicates are divided into the pseudo-data vectors and for each pseudo-data vector a sample from the posterior distribution is generated based on Equation 3.10.

3.2.3 Differential expression evaluation

We presented a probabilistic model of differential transcript expression and the inference for this model. By applying the model to expression estimates of multiple conditions we infer condition mean expression of transcripts of every condition, $\mu_m^{(c)} = \{\mu_m^{(c)(s)} | s = 1 \dots S\}$. By comparing the distributions of $\mu_m^{(c)}$ in different conditions we can draw conclusions about expression changes of transcript m.

Firstly, we compute the PPLR introduced at the beginning of this chapter. The probability of up-regulation of transcript m in first condition can be expressed

in terms of the PPLR,

$$PPLR_m = \frac{1}{S}\sum_{s=1}^{S} \delta\left(\mu_m^{(c1)(s)} > \mu_m^{(c2)(s)}\right). \tag{3.11}$$

Conversely, the probability of down-regulation is just $1 - PPLR_m$.

Subsequently we can rank the transcripts based on their estimated PPLR values. Setting a threshold α_{sig}, we can select all transcripts with $PPLR_m > 1 - \alpha_{sig}$ and $PPLR_m < \alpha_{sig}$ as being differentially expressed.

Apart from the evaluation of significance of DE, researchers are also interested in changes of expression represented by the log fold change FC_m. We can similarly estimate the expected log fold change between two conditions using the samples of condition mean expression,

$$\mathrm{E}[FC_m] = \frac{1}{\log(2)}\frac{1}{S}\sum_{s=1}^{S} \mu_m^{(c1)(s)} - \mu_m^{(c2)(s)}. \tag{3.12}$$

The condition mean expression is already in log-scale, thus for calculating the fold change we can simply subtract the estimates. We divide the result by log(2) as the log fold change is usually reported using base 2 logarithm.

3.3 Normalising expression from multiple experiments

In order to perform differential expression analysis, the RNA-seq samples should be sequenced with equivalent sequencing depth. This can be problematic as various conditions and replicates might be sequenced with changes in the preparation protocol, at different times and even in different laboratories. To make these kind of samples comparable, the expression estimates have to be normalised.

Expression levels reported in the RPKM measure are already normalised by the total sequencing output. This kind of normalisation is not always sufficient (Bullard et al., 2010). A large increase in expression level of a long transcript can affect the read distribution of the sequencing output. As more reads will be generated from one transcript, the relative proportions of other transcripts will decrease despite their constant expression.

The previously proposed normalisation methods were all applied to gene expression measured in read counts. Bullard et al. (2010) proposed normalisation procedure using quantile based normalisation, where the read count is normalised by the upper-quartile of the gene count. Robinson and Oshlack (2010) used Trimmed Mean of M-values (TMM), where M-value refers to the log ratio between two samples.

Both of these methods, as well as other approaches (Anders and Huber, 2010; Hansen et al., 2012), compare the distributions of read counts of genes and produce a multiplicative scaling factor for adjusting the total read count in each experiment. This makes the approaches universally interchangeable and enables us to use them in our differential expression model as well.

Applying normalisation methods

The proposed differential expression model (Figure 3.2) uses log expression samples measured in RPKM as an input. The model includes sample-specific normalisation constants $n^{(rc)}$ for each replicate of each condition. In order to use one of the existing methods, we first have to convert the mean expression levels of each replicate into estimated counts. The methods output a scaling factor for library size of each replicate, $s^{(cr)}$. To apply the factor to expression estimates in RPKM, we have to divide the RPKM expression by the scaling factor. Hence the normalisation constant used in our model is calculated as

$$n^{(cr)} = -\log s^{(cr)}, \tag{3.13}$$

and is used as an additive term, since the model works in logarithmic space.

We also have to consider the fact that the original methods were devised for gene counts normalisation, whereas our model works with transcript expression level. While we can use counts to represent transcript expression, the higher variability of transcript expression might affect the use of some of these methods, which usually rely on stable expression of most of the transcripts. An alternative is to calculate per-gene read counts and subsequently use them as an input for normalisation method to calculate the scaling factor. As the scaling factor applies to the library size it can be applied to both transcript and gene expression.

3.4 Results

We apply the proposed probabilistic inference to both real and synthetic RNA-seq datasets. While we can use real data from the microRNA target study to demonstrate the use of our method, there is no way of validating the results. We use synthetic data to evaluate the performance of our approach.

In each case we analyse data consisting of two conditions with biological replicates in each condition. Individual replicates are analysed with transcript expression quantification method presented in Chapter 2. The quantification method draws samples from the posterior distribution of transcript expression, which are converted into Log RPKM and used as an input for our probabilistic DE analysis.

For the purpose of validation with synthetic data, we use an empirical approach for setting the initial ground truth expression of transcripts, with the emphasis on preserving realistic expression levels with realistic biological fluctuations within each condition. While the approach is dependent on measurements from real datasets, it provides independent ground for comparing our method with alternative DE approaches.

The DE analysis methodology is implemented in the BitSeq application for transcript expression inference and DE assessment.

3.4.1 Analysis of real data

DE analysis example on a single transcript

We use data from the study by Xu et al. (2010) to demonstrate the DE analysis procedure using our model. In the report, the authors study the transcriptomic targets regulated by micro-RNA miR-155. miR-155 is one of the most highly implicated microRNAs related to hematologic and other cancers (Xu et al., 2010). The study compares two conditions, one being Burkitt's lymphoma Mutu cell line infected with miR-155 retrovirus, while the other is Burkitt's lymphoma Mutu cell line infected with control retrovirus. The data was downloaded from the Short Read Archive (NCBI, 2010), accession number SRP001880.

We selected this dataset, because it contains technical and biological replication for both studied conditions. We observed significant difference between biological and technical variance of expression estimates, see Figure 2.13 presented earlier. Furthermore, the prominence of biological variance increases with

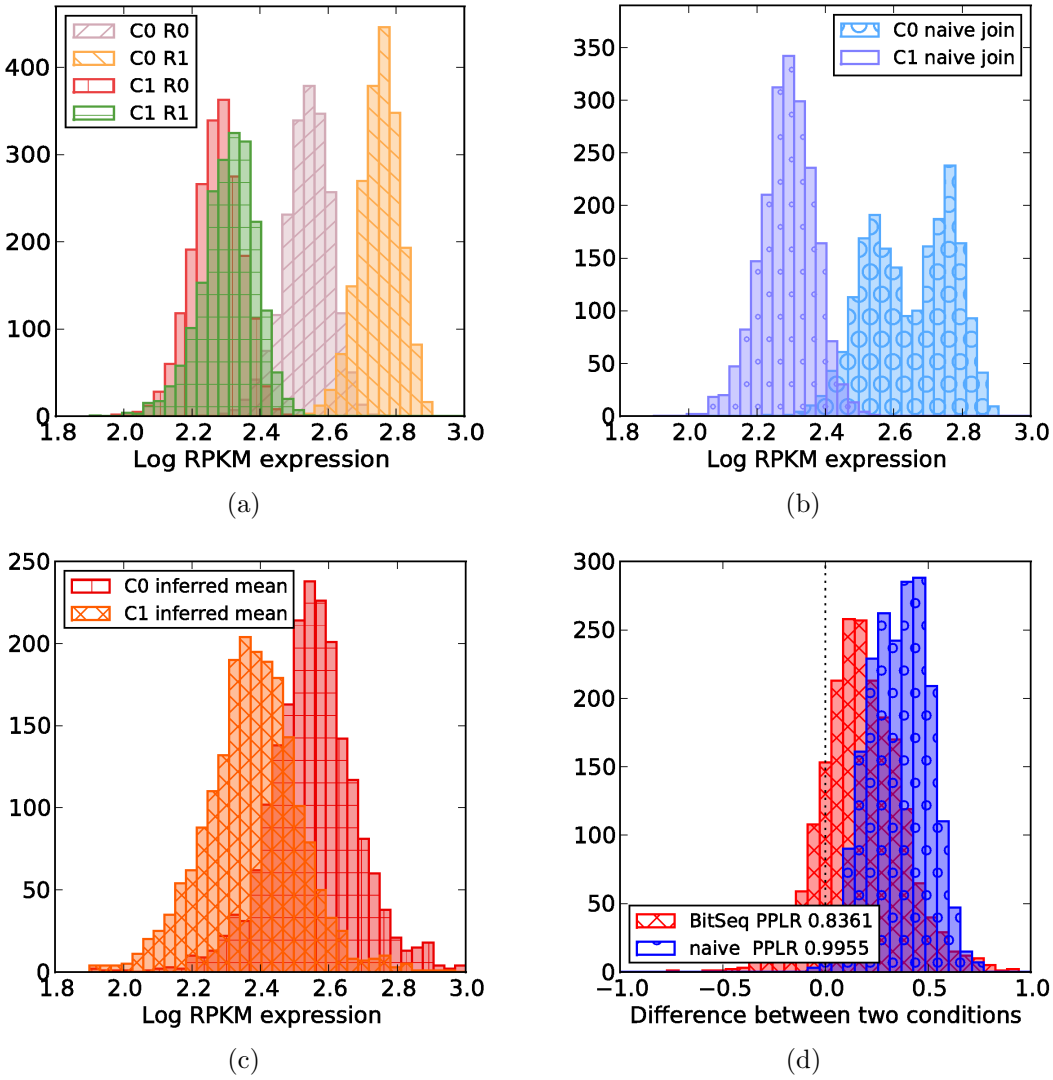


Figure 3.3: **Comparison of the DE model to naive approach for combining replicates within a condition.** Single transcript, uc001avk.2, of the Xu et al. (2010) dataset is assessed. (a) Initial posterior distributions of transcript expression levels for two conditions (labeled C0, C1), with two biological replicates each (labeled R0, R1). (b) Mean expression level for each condition using the naive approach for combining replicates. The posterior distributions from replicates are joined into one dataset for each condition. (c) Inferred posterior distribution of mean expression level for each condition using our probabilistic model. (d) Distribution of differences between conditions from both approaches show that the naive approach leads to overconfident conclusion.

transcript expression level. We illustrate how BitSeq handles biological replicates to account for this variance in Figure 3.3, by showing the modelling process for one example transcript given only two biological replicates for each of two conditions.

Figure 3.3(a) shows histograms of expression level samples produced in the first stage of our pipeline. BitSeq probabilistically infers condition mean expression levels using all replicates. For comparison, we used a naive way of combining two replicates by combining the posterior distributions of expression into a single distribution. The resulting posterior distributions for both approaches are depicted in Figures 3.3(b) and 3.3(c).

The probability of differential expression for each transcript is assessed by computing the difference in posterior expression distributions of the two conditions. Resulting distributions of differences for both approaches are portrayed in Figure 3.3(d) with obvious difference in the level of confidence. The naive approach reports high confidence of up-regulation in the second condition, with the probability of positive log ratio (PPLR) being 0.995. When biological variance is being considered by inferring the condition mean expression, the significance of differential expression is decreased to PPLR 0.836.

3.4.2 Evaluation on synthetic data

Unlike the problem of transcript expression quantification, realistic evaluation of DE analysis performance is difficult. There are no well established RNA-seq datasets that would contain at least two conditions with biological replicates, having known or validated differentially expressed transcripts. Therefore, we have to use synthetic data to evaluate and compare various methods.

In the transcript expression simulation, we are simulating known procedure of high-throughput sequencing. On the other hand, the properties which are important for DE analysis are unknown. These are the natural fluctuations within conditions and true changes of abundance between conditions. While it is possible to measure these properties with sequencing or alternative technologies, they depend on specific conditions and transcripts, making their replication very difficult.

Here we use our own simulation procedure based on observed expression levels and variations in biological replicates.

Simulating differential expressed data

We created an empirical method for simulating datasets with two conditions and biological replicates based on the estimated expression levels from Xu et al. (2010) dataset. The aim of the procedure is to generate ground truth transcript expression levels for each replicate of each condition. We define the differential expression in terms of changes in transcript abundance instead of considering just entire genes. Once the ground truth transcript expression is set, we can generate reads with an arbitrary RNA-seq read simulator.

The synthetic data is generated with transcripts having the same mean expression and biological variance as was observed in the real data. We use the mean expression over all replicates of all conditions to set the initial transcript expression. Subsequently, we select a certain fraction of transcripts to be differentially expressed. These transcripts are chosen randomly and their expression is increased or decreased by a fold change, which is either fixed or selected randomly from a pre-defined range. This defines the base expression for each condition. As we also have to create the biological replicates, we use the observed differences of transcript expression between replicates of each condition to simulate expression changes within each condition.

Using the ground truth expression levels, we generate RNA-seq reads using our own simulation procedure described in Section 2.4.5. However, an alternative RNA-seq simulation could have been used in similar fashion.

Here we use only 7537 transcripts from chromosome 1 to enable a smaller dataset and more efficient analysis, allowing for multiple validation runs. Approximately a third of transcripts are set to be DE in each run, with half being up-regulated and half being down-regulated. After adding the biological variations to the base expression levels, we simulated 500K single-end reads from each replicate.

Evaluation

Using artificially simulated data with a predefined set of differentially expressed transcripts, we evaluated our approach and compared it with four other methods commonly used for differential expression analysis. DESeq v1.6.1 (Anders and Huber, 2010), edgeR v2.4.3 (Robinson et al., 2010), baySeq v1.8.1 (Hardcastle and Kelly, 2010) were designed to operate on the gene level and Cuffdiff v1.3.0(Trapnell et al., 2010) on the transcript level. Despite not being designed

for this purpose, we also consider the first three gene-level methods in this comparison as the use case is very similar and all are well established methods used for DE. We used transcript expression estimates from BitSeq stage 1 as an input for all methods except Cuffdiff, which has its own expression quantification pipeline based on TopHat and Cufflinks. We converted the relative expression of fragments into the expected read counts by simply multiplying it by the total number of aligned reads. We used default settings for each of the methods according to the provided manual or vignette.

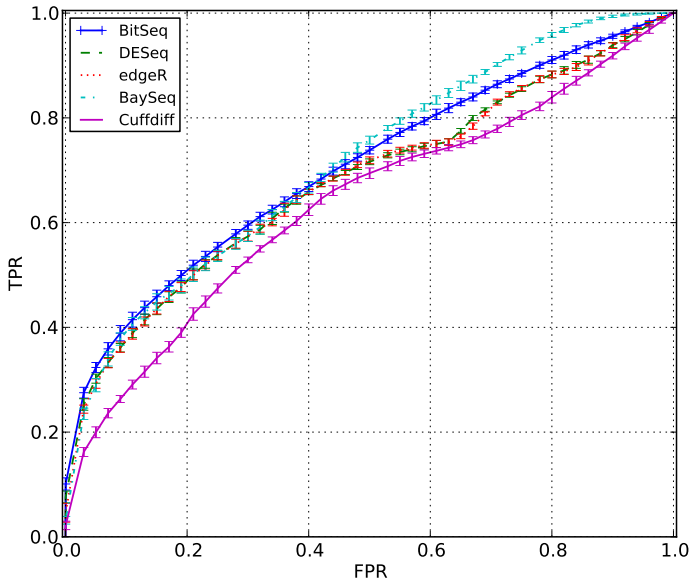
The methods are compared using ROC plots based on the known DE transcripts. Figure 3.4(a) shows the overall results of the five methods generated by averaging over 5 runs. Only transcripts with at least one generated read per replicate on average were included in individual comparisons. All runs use the same mean expression and biological fluctuations, which were generated by the above mentioned procedure. The runs differ in the choice of DE transcripts and the respective expression fold change which was selected randomly from the interval $(1.5, 3.5)$.

We have also included Precision-Recall plot of the same results in Figure 3.4(b). Here precision, or positive predictive value, is plotted against recall, or true positive rate.

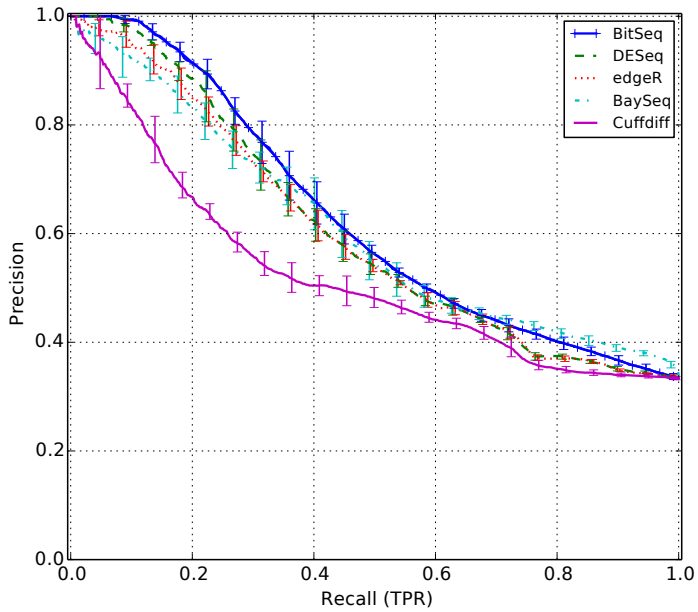
From both plots we can see that BitSeq is the most accurate method; baySeq, edgeR and DESeq provide comparable level of accuracy; with Cuffdiff further behind. In terms of DE analysis, we are usually interested only in the results with low false positive rate. Hence we focus our further ROC comparisons on regions up to 0.2 FPR, which are relevant for DE experiments.

We further compare the effectiveness of the methods with respect to the expression level. Figure 3.5 shows the same results with transcripts split into three groups based on the pre-set expression levels. BitSeq's advantage is especially clear for lower expression levels (Figures 3.5(a), 3.5(b)). The overall performance here is fairly low, because of high level of biological variance. For highest expressed transcripts (Figure 3.5(c)), DESeq and edgeR show slightly higher true positive rate than BitSeq and baySeq, especially at larger false positive rates.

In the last comparison presented in Figure 3.6, we compare the accuracy of these methods with respect to the fold change of differentially expressed transcripts. We again restrict the figures to the area with false positive rate below 0.2 which in our opinion is the most important in terms of applicability. Instead

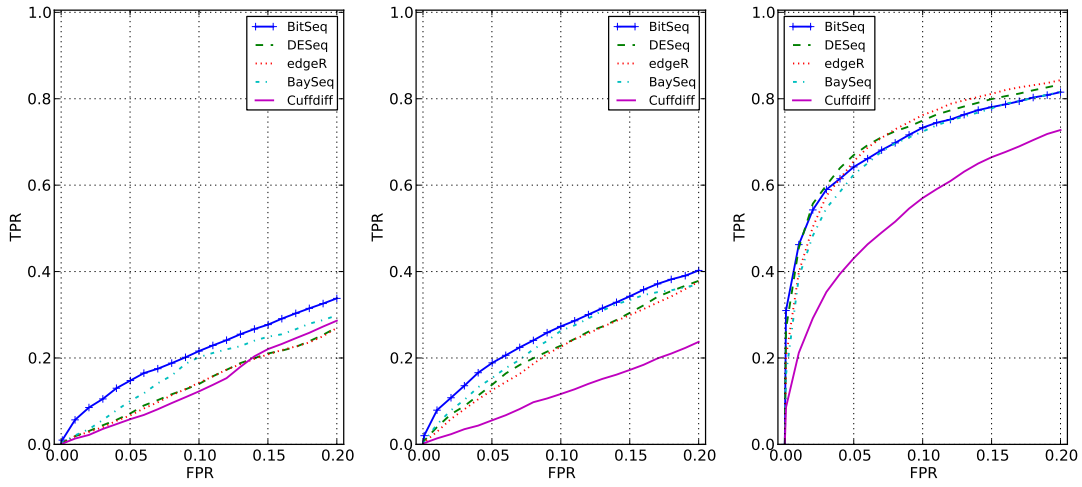


(a)



(b)

Figure 3.4: **Evaluation of transcript level DE analysis using artificial dataset, comparing BitSeq with alternative approaches.** (a) ROC curves averaged over 5 runs with different sets of DE transcripts; standard deviation depicted by error bars. (b) Precision-Recall curves for the same results.



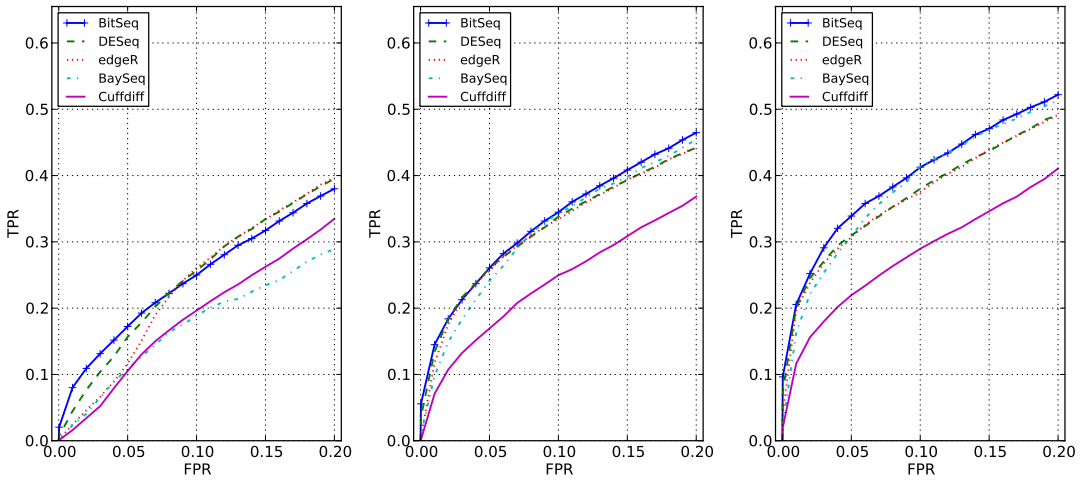
(a) (b) (c)

Figure 3.5: **DE performance comparison with respect to varying expression levels.** The curves are averaged over 5 runs with different set of transcripts being differentially expressed by fold change uniformly distributed in the interval $(1.5, 3.5)$. We discarded transcripts without any reads initially generated as these provide no signal. Transcripts were divided into 3 equally sized groups based on the mean generative read count: (a) $[0, 3)$, (b) $[3, 19)$ and (c) $[19, \infty)$.

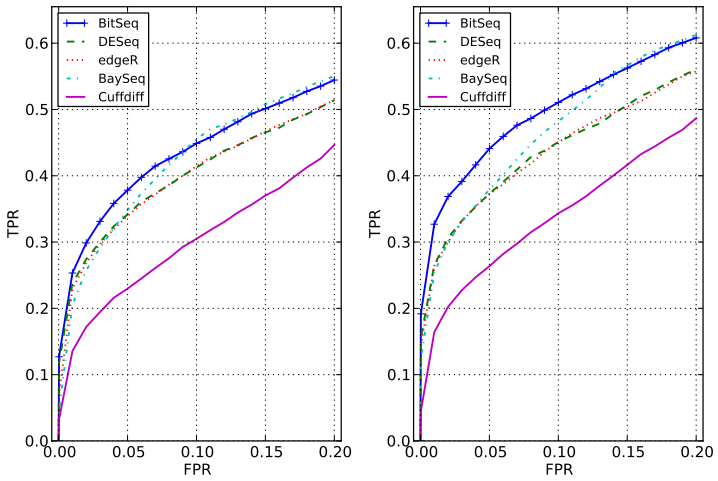
of using randomly selected fold change, all differentially expressed transcripts are either up-regulated or down-regulated by a constant fold change. The increase of fold change clearly improves the performance of the methods as we expected. BitSeq and baySeq have consistently better results than the other methods except for the lowest fold change 1.5, in which baySeq has the lowest true positive rate and edgeR with DESeq outperform BitSeq in half of the spectrum.

In all of our DE experiments, Cuffdiff, despite being designed for transcript level analysis, performs worse out of the 5 compared algorithm. Our data also shows that for most parts, the DESeq and edgeR methods produce very similar results in terms of accuracy. This could be explained by the fact that the methods use similar model based on negative binomial distribution with the same inference procedure, while having different approach of accounting for the biological variance.

We have to note, that even though we tried to simulate the data in way to resemble real RNA-seq experiments, the DE analysis proved to be rather difficult for all methods being compared. A possible cause for this could be high biological



(a) (b) (c)



(d) (e)

Figure 3.6: **DE performance comparison with respect to various levels of expression fold change.** The figures focus on the most relevant region with false positive rate below 0.2, and showing the y-axis up to true positive rate 0.65. The sub figures show results for varying expression fold change: 1.5 (a), 2.0 (b), 2.5 (c), 3.0 (d) and 5.0 (e).

	Accept null	Accept alternative	Sum
Null true	U	V	m_0
Alternative true	T	S	m_1
Sum	W	R	m

Table 3.1: **Multiple testing outcomes.** The possible outcomes of performing m multiple hypothesis tests. S is the number of true positives; V is the number of false positives, or Type I error; and T is the number of false negatives, or Type II error.

variance within replicates or poor read coverage of some of the transcripts.

3.4.3 Estimating False Discovery Rate

The differential expression analysis method uses multiple hypothesis testing to evaluate the probability of DE for individual transcripts and genes. In multiple hypothesis testing it is often useful to control the rate of false positives within the results. Moreover, in many studies, the significance threshold is determined based on an acceptable proportion of false positives. We applied the positive False Discovery Rate (pFDR) method for controlling the number of false positive proposed by Storey (2002). The pFDR is an alternative formulation to the False Discovery Rate (FDR) proposed by Benjamini and Hochberg (1995) often used in traditional frequentist hypothesis testing. Storey (2003) argues for the use of the pFDR instead and provides Bayesian interpretation of the multiple hypothesis testing error.

Given a significance region Γ and possible outcomes described in Table 3.1 the pFDR is defined as

$$\text{pFDR}(\Gamma) = \text{E}\left[\frac{V(\Gamma)}{R(\Gamma)}\middle| R(\Gamma) > 0\right]. \tag{3.14}$$

Storey (2003) showed that for test statistics $T_1, \ldots, T_M$ and hypotheses $H_1, \ldots, H_M$ assuming that random variables (T_i, H_i) are i.i.d., the pFDR can be estimated by

$$\text{pFDR}(\Gamma) = P(H = 0 | T \in \Gamma) = \sum_{i; T_i \in \Gamma} \text{E}[\delta(H_i = 0)]. \tag{3.15}$$

He further defined a q-value, an analogue to the p-value used for determining the

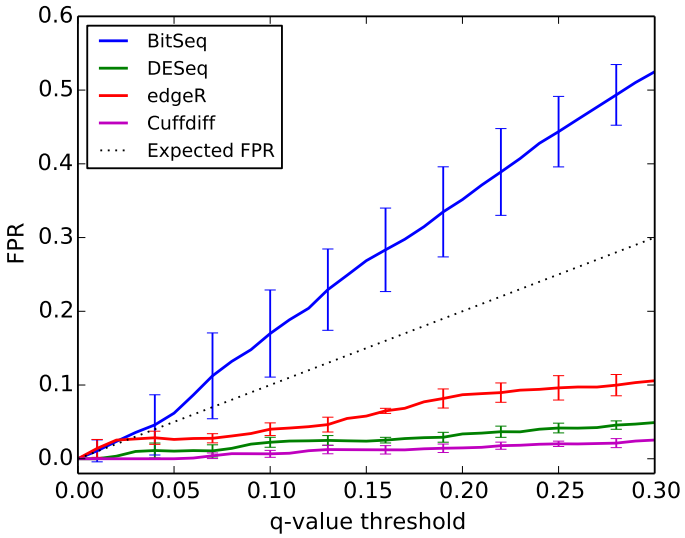


Figure 3.7: **False positive rate evaluation on synthetic dataset.** Figure shows true FPR of transcripts selected based on the q-value. The value is averaged over 5 runs, using standard deviation as errorbars.

FDR. The q-value for test statistic t is defined as

$$q(t) = \inf_{\{\Gamma; t \in \Gamma\}} \text{pFDR}(\Gamma), \tag{3.16}$$

hence the q-value estimates the strength of the test statistic t with respect to the pFDR (Storey, 2003).

In terms of DE analysis, the null hypothesis means no change of expression, while the alternative hypothesis signifies differentially expressed gene or transcript. In our case, the output is the PPLR, a one-sided test for up-regulation, which cannot be used directly for controlling the pFDR of DE testing. To control the pFDR for DE, we calculate q-value for both PPLR and inverse PPLR $iPPLR_m = 1 - PPLR_m$, which denotes the probability of down-regulation. For a desired pFDR threshold α, we select all transcripts with either of the q-values below α.

We applied this procedure to synthetic data generated by the procedure presented in previous section. As we know which transcripts are truly DE, we can calculate the actual proportion of false positives between transcripts that were selected as DE by our method. For comparison, we include DESeq, edgeR and Cuffdiff, which also provide a q-value for controlling the proportion of false discoveries.

Results of the evaluation are presented in Figure 3.7. We can see that BitSeq underestimates the q-value with respect to the true rate of false positives. On the other hand, all other methods provide very conservative estimates of the false positive rate and hence decrease the potential power of the DE test. In this case, the use of both our and alternative method's q-values for selecting significant DE transcripts is questionable.

The underestimation of the actual false positive rate can be caused by the fact that the DE tests are not fully independent. While the final stage of DE analysis applies the model to individual transcripts, the transcripts share the expression dependent hyperparameters. Furthermore, as we have shown earlier, the expression levels can be highly correlated. Wrong quantification of one transcript affects other transcripts within a gene, causing correlated false positive calls for transcripts of a single gene.

3.5 Summary

In this chapter we have presented a probabilistic inference method for differential expression analysis of transcripts and genes. Input for the method are marginal posterior distributions of individual transcript expression levels. We use the expression inferred by our transcript expression quantification method presented in Chapter 2, however, other methods that could output a full posterior distribution can be used instead.

The probabilistic DE model is based on a Log-Normal model of observed expression and accounts for biological variance through the use of expression estimates from multiple biological replicates. Instead of assuming a certain parametric form of the within-sample transcript expression distribution, we apply the model to pseudo-data vectors of individual samples across replicates. Applying the model to the pseudo-data vectors of MCMC samples yields samples from the posterior distribution of condition mean expression for each condition. We use the probability of positive log ratio to test for up-regulation and down-regulation of individual transcripts, producing ranking of the most likely DE transcripts.

We evaluated the method on synthetic data with a known set of differentially expressed transcripts. In comparison with other alternative approaches, our method outperforms transcript level DE tool Cuffdiff as well as methods designed for gene DE analysis. Our method shows greater accuracy especially in

case of low and medium expressed transcripts.

We have examined the use of pFDR method for estimating the false positive rate for a certain significance threshold. The pFDR in our case does not seem to be well calibrated and requires further study of advanced approaches accounting for correlated transcripts. Note that the other methods included in our comparison also showed poorly calibrated estimates of false positive rate.

Chapter 4

Applying deterministic approximate inference methods

In Chapter 2 we presented a Bayesian probabilistic model for transcript expression quantification. We applied the Markov chain Monte Carlo algorithm to infer the posterior distribution of expression, as the exact form of the distribution is not analytically tractable. The MCMC algorithm, once it has converged, enables us to sample from the true posterior distribution without knowing the marginal likelihood. However for large datasets, the time complexity of the algorithm becomes its major drawback.

The size of RNA-seq experiments continually increases with improvements in the sequencing technology. In the MCMC algorithm, the number of samples necessary to ensure the convergence of the Markov chains and to capture the full posterior increases as the number of reads grows and makes the problem more complex. Furthermore, the time complexity of generating one sample scales linearly with the number of alignments which is directly related to the number of sequenced reads. The constant growth of sequencing datasets escalates the need for more efficient inference methods within the Bayesian framework.

Here we present a Variational Bayes inference approach for transcript expression quantification. The Variational Bayes approach is a deterministic approximate inference method which, in most cases, provides computationally more efficient inference for Bayesian probabilistic models. In the Variational Bayes inference approach the posterior probability distribution is approximated by a tractable distribution q which can be usually derived from the model by assuming additional independence between parameters. The distribution q is then

optimised in terms of minimisation of the divergence from the true posterior distribution.

This chapter is a result of collaboration with James Hensman. Methodology of the Variational Bayes inference algorithm presented in this chapter was proposed and derived by James Hensman and combines the previous model proposed by Peter Glaus presented in Chapter 2 and collapsed Variational Bayes with conjugate gradient optimisation approach presented in Hensman et al. (2012). The algorithm, with further computational optimisations and parallelisation was implemented by Peter Glaus, who also carried out the experiments and comparisons presented in Section 4.2.

4.1 Variational Bayes inference

The inference method presented here is based on the probabilistic generative model of RNA-seq outlined in Figure 2.2. The model can be described in terms of a mixture model in which data is derived from a mixture of different transcripts, the mixture components, with each read originating from one component. Each component is defined by a specific probability distribution over the data it generates. Although reads originate from only one component they may map to multiple related components, resulting in some ambiguity in their assignment. Transcript expression levels are model parameters (mixture component proportions) that have to be inferred from the mapped read data. Due to their probabilistic nature, the mixture models can fully account for multiple mapping reads, complex biases in the sequence data, sequencing errors, alignment quality scores and prior information on the insert length in paired-end reads.

The probabilistic model is shown using standard directed graphical notation in Figure 4.1. To facilitate the derivations the model is simplified through exclusion of the noise-specific parameters θ^{act} and Z_n^{act}. Instead a *noise* transcript is added and treated as one of the known transcripts. Here θ_0 is equivalent of the noise parameter θ^{act} and represents the proportion of reads that could not have been assigned to any known transcript with enough certainty. As the marginal univariate distribution of the Dirichlet distribution is the Beta distribution (used for θ^{act}) and the model variables are conjugate, the model is the same as that in Chapter 2 used with MCMC, subject to a slight reformulation of the prior parameters.

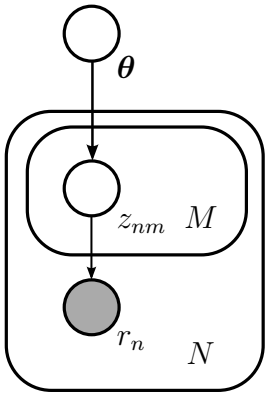


Figure 4.1: **Graphical model of the RNA-seq mixture problem used in Variational Bayes inference.** Given a known transcriptome and observed reads R, the inference problem is for $\boldsymbol{\theta}$ through the latent variables $\mathbf{Z}$.

We further replaced the indicator vector $I_n \in \{0, \ldots, M\}; n = 1 \ldots N$ that assigns reads to transcripts with a binary indicator matrix $z_{nm} \in \{0, 1\}$, which is more common in the mixture model literature. Here $\mathbf{z}_n$ is the allocation vector of read r_n and $\mathbf{Z}$ denotes the collection of all allocations. $\mathbf{T}$ denotes the set of transcripts, using T_m to refer to transcript m and $P(r_n \,|\, T_m) = P(r_n \,|\, I_n = m)$ to denote the likelihood of read r_n originating from transcript m as defined in Section 2.2.

We focus on the mixture part of the analysis, assuming that the model which associates reads to transcripts, $P(r_n \,|\, T_m)$, is known. Following the approach from Chapter 2, we compute this part of the model *a priori*, with parameters estimated from uniquely aligned reads.

4.1.1 The generative model

The generative model for an RNA-seq assay is as follows. We assume that the experiment consists of a pile of RNA fragments, where the abundance of fragments from transcript m in the assay is θ_m. Fragments are then sequenced in these proportions, so that the prior probability of any fragment corresponding to transcript m is θ_m. Introducing the allocation vector $\mathbf{z}_n$ for each read, we can write

$$P(\mathbf{Z}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \theta_k^{z_{nm}}, \tag{4.1}$$

where $z_{nm} \in \{0, 1\}$ is a binary variable which indicates whether the n^{th} fragment came from the m^{th} transcript ($z_{nm} = 1$) and is subject to $\sum_{m=0}^{M} z_{nm} = 1$.

We note that both $\boldsymbol{\theta}$ and $\mathbf{Z}$ are variables to be inferred, with $\boldsymbol{\theta}$ the main object of interest as it is the relative proportion of transcripts' fragments, which can be transformed into alternative expression measure as described in Section 1.4.2. The variables $\mathbf{Z}$ are latent variables, whilst they are not of interest directly, inference of these variables is essential in order to infer $\boldsymbol{\theta}$.

The final part of our model is to specify a prior belief in the vector $\boldsymbol{\theta}$. To make our approximations tractable, we again use a conjugate prior, which in this case is a Dirichlet distribution,

$$P(\boldsymbol{\theta}) \sim \text{Dir}(\boldsymbol{\alpha}^o), \tag{4.2}$$

where α_m^o represents our prior belief in the values of θ_m and $\hat{\alpha}^o = \sum_{m=1}^{M} \alpha_m^o$. We use a weak but proper prior $\alpha_m^o = 1$; $m = 0 \ldots M$. A priori, we assume that the concentrations are all equal, but with large uncertainty.

4.1.2 Approximate inference

We are interested in computing the posterior distribution for the mixing proportions, $P(\boldsymbol{\theta} \,|\, R, \mathbf{T}) \propto \sum_{\mathbf{Z}} P(R \,|\, \mathbf{T}, \mathbf{Z}) P(\mathbf{Z} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta})$. Variational Bayes involves approximating the posterior probability density of all the model parameters with another distribution q,

$$q(\boldsymbol{\theta}, \mathbf{Z}) \approx P(\boldsymbol{\theta}, \mathbf{Z} | R, \mathbf{T}). \tag{4.3}$$

The approximation is optimised by minimising the Kullback-Leibler (KL) divergence between $q(\boldsymbol{\theta}, \mathbf{Z})$ and $P(\boldsymbol{\theta}, \mathbf{Z} | R, \mathbf{T})$. To make the VB approach tractable, some factorisations need to be assumed in the approximate posterior. In the case of the current model, we assume that the posterior probability of the transcript proportions factorises from the alignments:

$$q(\boldsymbol{\theta}, \mathbf{Z}) = q(\boldsymbol{\theta}) q(\mathbf{Z}). \tag{4.4}$$

Further factorisations in $q(\mathbf{Z})$ occur due to the simplicity of the model, revealing $q(\mathbf{Z}) = \prod_{n=1}^{N} q(\mathbf{z}_n)$.

We write the approximate distribution for $q(\mathbf{Z})$ using the parameters ϕ_{nm},

which denotes the approximate posterior probability of $z_{nm} = 1$:

$$q(\mathbf{Z}) = \prod_{n=1}^{N} \prod_{m=1}^{M} \phi_{nm}^{z_{nm}}. \tag{4.5}$$

We need not introduce parameters for $q(\boldsymbol{\theta})$ since it will arise implicitly in our derivation in terms of ϕ.

The objective function

Approximate inference is performed by optimisation: the parameters of the approximating distribution are changed so as to minimise the KL divergence. Whilst the KL divergence is not computable, it is possible to derive a lower bound on the marginal likelihood, maximisation of which minimises the KL divergence (see e.g. Bishop, 2006). Here we derive a *collapsed* lower bound which is dependent only on the parameters of $q(\mathbf{Z})$, with the optimal distribution for $q(\boldsymbol{\theta})$ arising implicitly for any given $q(\mathbf{Z})$.

First we construct a lower bound on the conditional log probability of the reads R given the transcript concentrations $\boldsymbol{\theta}$ and the known transcriptome $\mathbf{T}$:

$$\begin{aligned} \ln P(R \,|\, \mathbf{T}, \boldsymbol{\theta}) &= \ln \int P(R \,|\, \mathbf{Z}, \mathbf{T}) P(\mathbf{Z} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z} \\ &\geq \mathbb{E}_{q(\mathbf{Z})} \Big[\ln P(R \,|\, \mathbf{Z}, \mathbf{T}) + \ln P(\mathbf{Z} \,|\, \boldsymbol{\theta}) - \ln q(\mathbf{Z}) \Big] \\ &\geq \sum_{n=1}^{N} \sum_{m=1}^{M} \phi_{nm} \big(\ln P(r_n \,|\, T_m) + \ln \theta_m - \ln \phi_{nm} \big) \\ &= \mathcal{L}_1(\boldsymbol{\theta}), \end{aligned} \tag{4.6}$$

where the first line follows from Jensen's inequality in a similar fashion to standard VB methods. We have denoted this conditional bound $\mathcal{L}_1(\boldsymbol{\theta})$, which is still a function of $\boldsymbol{\theta}$. In order to generate a bound on the marginal likelihood, $P(R \,|\, \mathbf{T})$, we need to remove this dependence on $\boldsymbol{\theta}$ which we do in a Bayesian fashion, by substituting $\mathcal{L}_1(\boldsymbol{\theta})$ into the following Bayesian marginalisation:

$$\begin{aligned} P(R \,|\, \mathbf{T}) &= \int P(R \,|\, \mathbf{T}, \boldsymbol{\theta}) P(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \\ &\geq \int \exp\{\mathcal{L}_1(\boldsymbol{\theta})\} P(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}. \end{aligned} \tag{4.7}$$

Solving this integral and taking the logarithm gives us our final bound which

equates to

$$\begin{aligned}\ln P(R \,|\, \mathbf{T}) \geq \mathcal{L} = \sum_{n=1}^{N}\sum_{m=1}^{M} \phi_{nm}\big(\ln P(r_n \,|\, T_m) - \ln \phi_{nm}\big) \\ + \ln \Gamma(\hat{\alpha}^o) - \ln \Gamma(\hat{\alpha}^o + N) - \sum_{m=1}^{M}\Big(\ln \Gamma(\alpha_m^o) - \ln \Gamma(\alpha_m^o + \hat{\phi}_m)\Big),\end{aligned} \tag{4.8}$$

where $\hat{\phi}_m = \sum_{n=1}^{N} \phi_{nm}$ and we also have that the approximate posterior distribution for $\boldsymbol{\theta}$ is a Dirichlet distribution with parameters $\alpha_m^o + \hat{\phi}_m$.

4.1.3 Optimisation

Having established the objective function as a lower bound on the marginal likelihood, all that remains is to optimise the variables of the approximating distribution $q(\mathbf{Z}, \boldsymbol{\theta})$. The dimensionality of this optimisation is rather high and potentially rather difficult. Optimisation in standard VB is usually performed by an EM like algorithm, which performs a series of convex optimisations in each of the factorised variables alternately. We refer to this procedure as VBEM or steepest gradient optimisation.

In our formulation of the problem, we only need to optimise the parameters of the distribution $q(\mathbf{Z})$, which we do by a gradient-based method. Taking a derivative of (4.8) with respect to the parameters ϕ gives

$$\frac{\partial \mathcal{L}}{\partial \phi_{nm}} = \ln P(r_n \,|\, T_m) - \ln \phi_{nm} + \psi(\alpha_m^o + \hat{\phi}_m), \tag{4.9}$$

where ψ is the digamma function. To avoid constrained optimisation we reparameterise ϕ as γ:

$$\phi_{nm} = \frac{e^{\gamma_{nm}}}{\sum_{m'=1}^{M} e^{\gamma_{nm'}}} \tag{4.10}$$

and it is then possible to optimise the variables γ using a standard gradient-based optimiser.

Geometry

Information geometry concerns the interpretation of statistical objects in a geometric fashion. Specifically, a class of probability distributions behaves as a Riemannian manifold with curvature given by the Fisher information. Amari

(1998) showed that the direction of the steepest descent on a such a manifold is given by the natural gradient:

$$\widetilde{\nabla}\mathcal{L} = G^{-1}\nabla\mathcal{L} \ , \tag{4.11}$$

where G is the Fisher information matrix. Since we are performing optimisation of the distribution $q(\mathbf{Z})$, we can make use of the natural gradient in computing a search direction. For our problem, we assume that the $N \times M$ matrix $\mathbf{Z}$ has been transformed into a NM vector, and the Fisher information corresponding to γ_{nm}, $\gamma_{n'm'}$ is given by

$$G[m,n,m',n'] = \begin{cases} \phi_{nm} - \phi_{nm}^2, & \text{if } n = n' \text{ and } m = m' \\ -\phi_{nm}\phi_{nm'}, & \text{if } n = n' \text{ but } m \neq m' \\ 0, & \text{otherwise.} \end{cases} \tag{4.12}$$

We note that this structure is block-diagonal, and that each block can be easily inverted using the Sherman-Morrison identity, giving an analytical expression for G^{-1}, and thus making the natural gradient very fast to compute. This differentiates our method from previous natural gradient-based methods for VB (Honkela et al., 2010), along with our use of the collapsed method.

The optimisation of the variational parameters then proceeds as follows. Following random initialisation, a unit step is taken in the natural gradient direction. Subsequent steps are subject to *conjugate* gradients (see Honkela et al., 2010). If the conjugate gradient step should fail to improve the objective we revert to a VBEM update, which is guaranteed to improve the bound. The conjugate gradient is computed either by Hestenes-Stiefel (HS) method (Hestenes and Stiefel, 1952) or by Fletcher-Reeves (FR) nonlinear conjugate gradient method (Fletcher and Reeves, 1964). For comparison of the two approaches, see results in Section 4.2.3.

Truncation

The optimisation described above has $N \times M$ free parameters for optimisation, one to align each read to each transcript. However, for most read-transcript pairs, $P(r_n \,|\, T_m)$ will be negligibly small. Similarly to the approach used in the MCMC inference, we truncate the values of $P(r_n \,|\, T_m)$ to zero if read r_n does not align to transcript m. Examining the objective function (4.8) we see that we can

also set ϕ_{nm} to zero for these truncated alignments (using the convention that $0 \ln(0) = 0$) and thus also $\gamma_{nm} = -\infty$ for the same. This truncation dramatically reduces the computational load of our algorithm, reducing the dimensionality of the optimisation space as well as reducing the number of operations needed to compute the objective.

4.1.4 The approximate posterior

Having fitted our model, we may wish to propagate the posterior distribution through a second set of processing, for example to identify differential expressed transcripts as in Chapter 3. Whilst it may be desirable to solve both stages together in a Bayesian framework, the size of the problem generally forbids this, therefore we propose the use of either a moment-matching or sampling procedure to propagate $q(\boldsymbol{\theta})$ through further analysis. The approximate posterior $q(\boldsymbol{\theta})$ is a Dirichlet distribution, whose marginals have the following useful properties:

$$\mathbb{E}\Big[\theta_m\Big] = \frac{\alpha^o_m + \hat{\phi}_m}{\hat{\alpha}^o + N}, \tag{4.13}$$

$$\text{var}[\theta_m] = (\alpha^o_m + \hat{\phi}_m)(\hat{\alpha}^o + N - \alpha^o_m - \hat{\phi}_m)C, \tag{4.14}$$

$$\text{cov}[\theta_m, \theta'_m] = -(\alpha^o_m + \hat{\phi}_m)(\alpha^o_{m'} + \hat{\phi}_{m'})C, \tag{4.15}$$

with $C = (\hat{\alpha}^o + N)^{-2}(\hat{\alpha}^o + N + 1)^{-1}$.

This approximate posterior is somewhat inflexible, in that it cannot express arbitrary covariances between the transcripts. This arises from the factorising assumption amongst the assignment of reads to transcripts: reads are assigned independently in the variational method and their dependence cannot be modelled. This is reflected in the results section where we show empirically that the VB approximation leads to an underestimation of the variance. Nonetheless, this simplifying assumption leads to reasonable levels of accuracy in terms of mean expression, and gives significant benefit in terms of speed increase. Note that most applications using expression level estimates only rely on the mean expression estimate without consideration of the full posterior.

	Measure (cutoff)			
Method	transcript (1)	relative (10)	relative (100)	gene (1)
BitSeq VB	0.994	0.941	0.961	0.994
BitSeq MCMC	0.994	0.945	0.963	0.994
TIGAR	0.998	0.944	0.963	0.999

Table 4.1: **The R^2 correlation coefficient of estimated expression levels and ground truth on synthetic data using VB inference.** Three different expression measures were used: absolute transcript expression, relative within-gene transcript expression and gene expression. Comparison includes sites with at least 1 read per transcript for transcript expression, either 10 or 100 reads per gene for within-gene transcript expression and at least 1 read per gene for gene expression.

4.2 Results and comparison with MCMC

We evaluate the accuracy of our inference approach using both synthetic and real data. The synthetic data enables comparison against known ground truth, whilst for the real data where the ground truth is unknown, we compare the VB method with a very long run of MCMC. We then return to the synthetic data in a comparison of differential expression analysis using the BitSeq pipeline.

4.2.1 Inference accuracy and performance on synthetic data

We use the same synthetic data as we have used in Section 2.4.5 to evaluate the accuracy of the MCMC method in comparison with three other transcript expression quantification tools. The expression is evaluated in three different measures: transcript expression, transcript within-gene relative proportions and gene expression. Here we additionally include TIGAR (Nariai et al., 2013), a recent method using generative probabilistic model with VB inference algorithm.

The Pearson R^2 correlation of the expression estimates with known expression levels are presented in Table 4.1 We see that in each measure, the variational approximation to the posterior performs almost as well as the MCMC implementation. TIGAR performs comparably to both BitSeq methods, though the differences are small.

On this relatively small dataset with 10 million simulated reads, the computational cost is significant for BitSeq and TIGAR (see Table 4.2). The MCMC

	Synthetic (10m reads)		Real (100m reads)	
Method	time (mins)	memory (GB)	time (mins)	memory (GB)
BitSeq VB	21	2.4	310	26.4
BitSeq MCMC	503	0.6	1769	8.5
TIGAR	509	8.2	n/a	∼80
Cufflinks	30	0.6	146	3.2

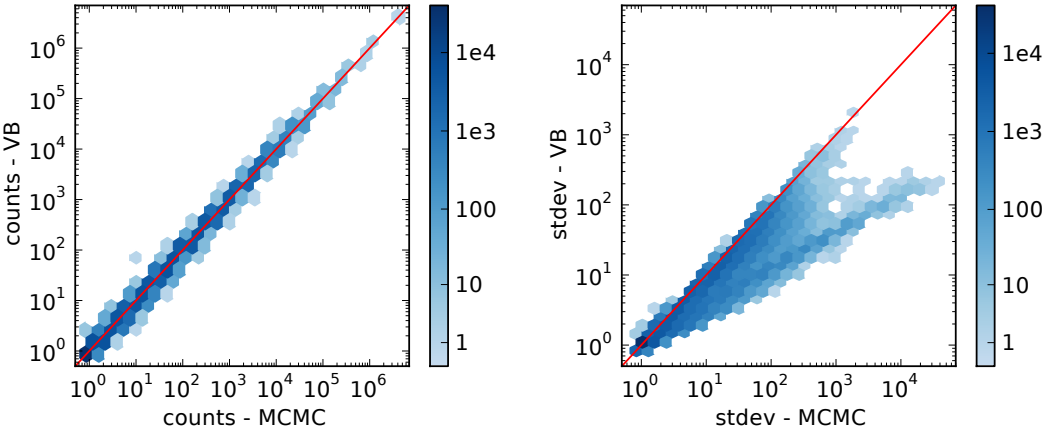
Table 4.2: **Comparison of run time and memory requirements for MCMC, VB and alternative VB implementation in TIGAR.** Smaller, synthetic, data was analysed on single CPU, while 4 CPUs were used for the real data consisting of 100m reads. Analysis was done on a computing node with Intel Xeon X5690, 3.47GHz CPU with 12.3MB cache.

version of BitSeq required 503 minutes, and TIGAR required 509 minutes. It is perhaps against the conventional wisdom that the Gibbs sampling procedure should be faster than TIGAR's variational method, and these differences may be due in part to the implementation, though we find the Gibbs procedure to be efficient in the next section also. We used single threaded mode for BitSeq as TIGAR does not provide explicit parallelisation option and seems to be using only one CPU.

The variational version of BitSeq, using the contemporary collapsed procedure defined above, takes significantly less time than either the BitSeq-MCMC method or TIGAR's VB at only 21 minutes. This represents a substantial difference that makes the approach attractive in circumstances where results are demanded quickly.

4.2.2 Analysis of RNA-seq data from the ENCODE project

We analyse RNA-seq reads downloaded from Short Read Archive (NCBI, 2010), experiment SRX110318, run SRR387661, generated by the ENCODE consortium (Djebali et al., 2012). Library extracted from cytosol of human bone marrow tissue affected by leukemia (K562) was sequenced by Illumina Genome Analyzer II, generating 124.8 million read pairs, 76 bp long. We mapped the reads using Bowtie 2.0.6 (Langmead and Salzberg, 2012) to a reference transcriptome using 140869 known coding sequences from Ensembl human cDNA, release 70 (Flicek et al., 2013). 98.8 million reads were mapped to the reference, with 5 mappings per read on average.



(a) Posterior mean transcript expression

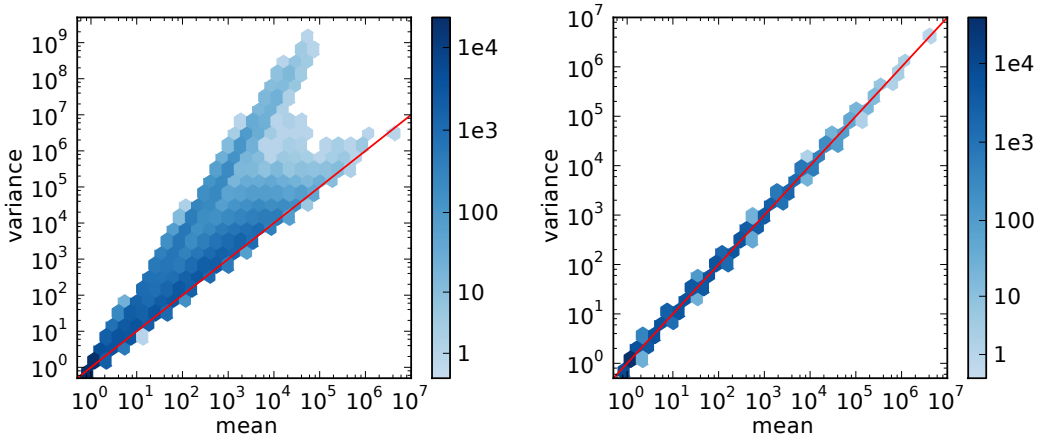
(b) Posterior standard deviation in transcript expression

Figure 4.2: **Comparison of the first two moments of the approximate posterior expression in counts per transcript.** The posterior distribution inferred by VB method is compared against the MCMC posterior distribution: (a) posterior mean (R^2 correlation is 0.999) (b) posterior standard deviation, the VB method significantly underestimates the posterior variance (σ^2).

Our main potential concern in using the variational method is the quality of approximation to the posterior. Figure 4.2 shows a comparison of the variational posterior with a ground truth computed by MCMC. We conclude that the VB method consistently provides very accurate estimates of the posterior mean across the whole range of expression levels. The Pearson R^2 correlation coefficient of mean expression levels of transcripts is 0.999. The estimates of posterior variance are less consistent: for a fraction of transcripts the variances are underestimated, sometimes rather severely.

We show the relationship between posterior mean and variance of transcript expression samples obtained by the two methods in Figure 4.3. We converted the inferred expression into estimated read counts per transcript. Here we can see that VB only estimates the Poisson variance of random sampling[1], which explains the underestimation of variance in comparison with MCMC which samples from the true posterior. Figure 4.3(a) shows that while expression estimates of some transcripts only exhibit the Poisson variance, the expression estimates of many transcripts vary more due to the multi-mapping reads.

[1]The variance of a Poisson distributed random variable is equal to the mean of the distribution.



(a) MCMC (b) VB

Figure 4.3: **Mean-variance relationship of the approximate posterior expression in counts per transcript.** (a) posterior distribution sampled by MCMC algorithm (b) Dirichlet distribution with parameters inferred by the VB algorithm.

The run time and memory requirements necessary for the analysis of this data are presented in Table 4.2. In this case, both inference methods were used in multi threaded setting facilitating 4 CPUs of the computing node with Intel Xeon 3.47GHz CPUs. Please note, both times include the same pre-processing stage which estimates likelihood for each alignment while accounting for non-uniform read distribution bias, which takes 162 minutes. If we subtract this time, then the actual convergence time for VB is significantly lower at 2.5 hours when compared to the collapsed MCMC at 26.8 hours. The memory requirements of our VB inference implementation were three times as high as for MCMC, but still proved feasible.

The memory requirements of TIGAR prohibited its use on this data set. Extrapolating linearly, we estimate that TIGAR would require 80GB of system memory to run, which is an infeasible resource for most practitioners. Indeed, Nariai et al. (2013) demonstrated their algorithm on data sets no larger than 4.5 million reads. For comparison, at the time of writing the Illumina website lists the HiSeq 2500 machine as capable of producing 3 billion reads in a single run.

We conclude that the novel variational method proposed here significantly outperforms the other methods in terms of computational time, and performs very well in estimating the mean of the posterior. If estimation of the expression level is all that is required, then it would seem that the VB method suffices.

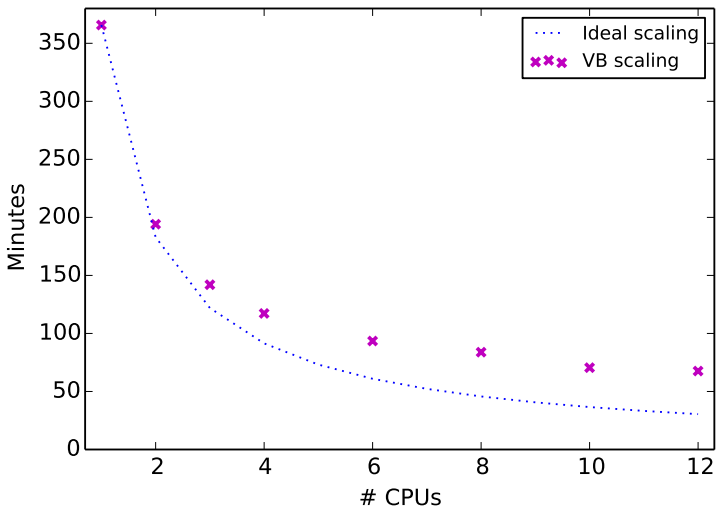


Figure 4.4: **Scaling of the parallelised VB inference.** We compare the run time of the VB inference on the K562 data with 98.8M mapped reads, with respect to the number of CPUs used. One through twelve Xeon 2.50GHz CPUs were used for assessment of the parallelisation efficiency. Perfect scaling curve is included for reference.

However, downstream methods which make use of uncertainty in the transcript quantification (such as the differential expression analysis proposed in BitSeq) may suffer from the poor approximation in terms of posterior variance.

Parallelisation

Our implementation of the Variational Bayes inference enables parallel computation of the natural gradients. We used the K562 data to assess the scaling of the parallelised computation on multiple CPUs. Here we used a computing node with two 6-core Xeon 2.50GHz CPUs with 15.4MB cache.

In Figure 4.4 we show the run time of the algorithm when using one through twelve CPUs. We include a curve for ideal case with perfect scaling calculated by dividing the run time for one CPU by the number of CPUs used. It is clear that using more than six CPUs provides minimal improvements, on the other hand, the use of more than one CPUs can dramatically decrease the run time of the algorithm.

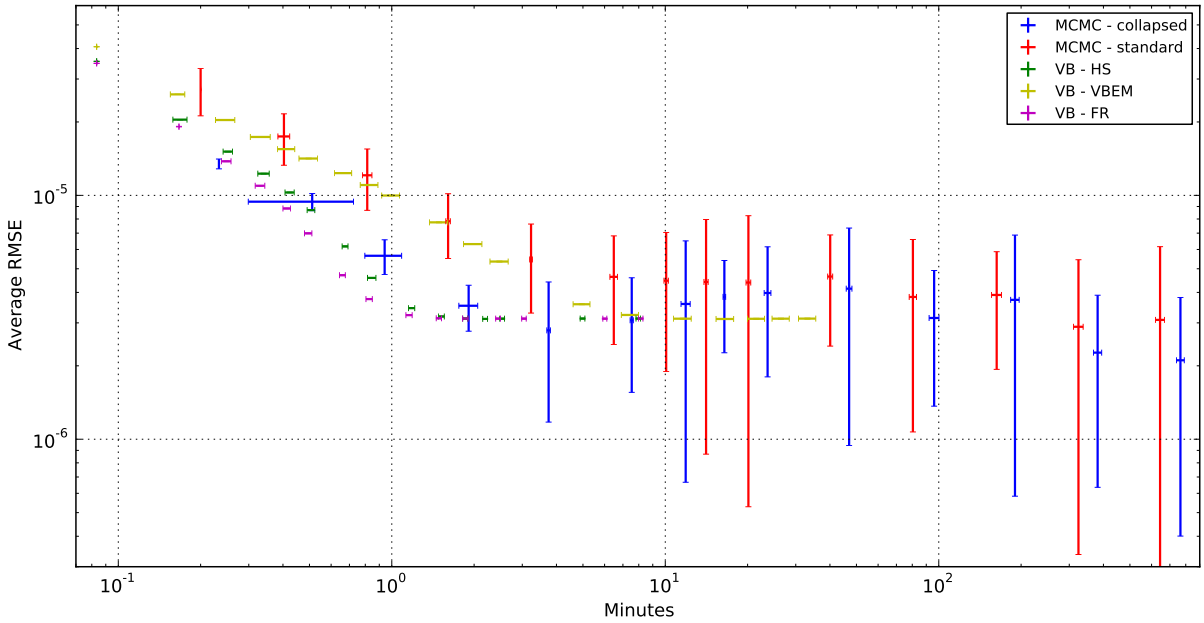


Figure 4.5: **Convergence comparison of MCMC and VB compared against long run of MCMC.** We compare collapsed Gibbs sampling, standard Gibbs sampling, VB with Hestenes-Stiefel optimisation, VB using steepest descent optimisation (VBEM) and VB with Fletcher-Reeves optimisation. Expression estimates obtained by a very long run of MCMC are used as a ground truth and the average root mean square error over 10 runs was calculated, two standard deviations are used as error bars. The VB methods showed negligible differences in convergence over several randomised initializations.

4.2.3 Convergence comparison

We further investigate convergence properties of MCMC and VB in terms of mean expression with respect to number of iterations and computational time. We use a subset of the data described in the previous section restricted to 8713 transcripts of chromosome 19. As the true expression is unknown, we use a long run of MCMC as the ground truth for mean expression estimates. Running the inference methods for certain number of iterations, we record the run time and calculate Root Mean Square Error (RMSE) of estimated expression.

The convergence of the variational methods and the Gibbs sampling procedures is shown in Figure 4.5. We compare collapsed Gibbs sampling, standard Gibbs sampling, VB with HS conjugate gradient optimisation, standard VB using steepest gradient ascent optimisation and VB with FR nonlinear conjugate gradient optimisation.

Our implementation of VB converges first in about 2 minutes. Surprisingly, some runs of collapsed MCMC converge to better estimates even faster than standard VB, which takes around 10 minutes. However, as MCMC is a stochastic method, an estimate that is consistently better than the results obtain by VB can be obtained after 900 minutes.

We can clearly see that the FR method outperforms the HS methods on this dataset. Furthermore, both conjugate gradient optimisations provide major improvement over the standard VBEM implementation.

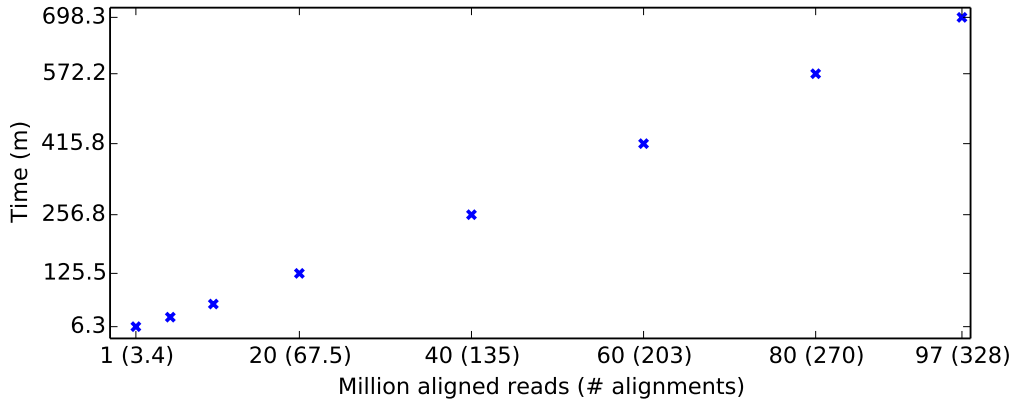
4.2.4 Efficiency of the VB inference with respect to sequencing depth

Here we look at the accuracy and efficiency of the VB algorithm when using different sized datasets. While we have shown high correlation of the mean expression estimates produced by VB and MCMC, we are interested whether this correlation depends on the number of reads within a dataset.

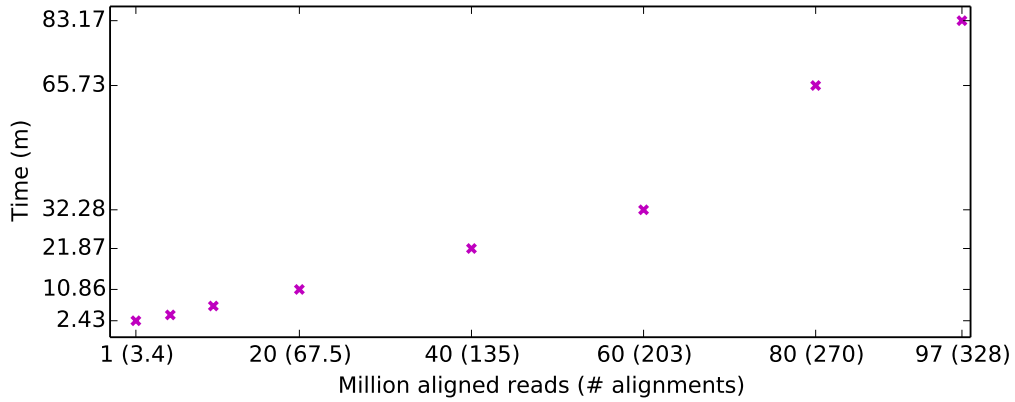
Here we use the same RNA-seq data, produced by the ENCODE consortium by sequencing K562 cell line, as we have used before. However, we have changed the preparation step by aligning to the Gencode reference sequence (Harrow et al., 2012), instead of the Ensembl reference. The Gencode reference is evidence-based and contains a more conservative set of only 94917 transcripts from 20720 genes. 97.0 million reads were mapped to the reference, with an average of 3.4 alignments per read. After computing the likelihoods for each alignments in the pre-processing step of our analysis, we sub-sampled the dataset. Given the 97 million mapped reads, we randomly sampled datasets consisting of 1, 5, 10, 20, 40, 60 and 80 million reads, simulating varying sequencing depth of the experiment. Excluded reads had all their alignments discarded, while we kept all alignments of the non-excluded reads.

We analysed the different sized datasets with both MCMC and VB approaches and compared the mean transcript expression estimates. For all 'sequencing depths' the Pearson R^2 correlation of the transcript mean expression estimates was above 0.9999.

We have also analysed the run time performance of the algorithms with respect to the number of reads. Using the sub-sampled data avoids differences due to various concentrations and biases, enabling objective comparison. In Figure 4.6



(a) MCMC



(b) VB

Figure 4.6: **Run time dependency of inference algorithms for various sizes of sequencing input.** We used four Xeon 2.50GHz CPUs to run the inference algorithms: (a) collapsed Gibbs sampler (b) Variational Bayes using FR conjugate gradient optimisation.

we show the run time of the two inference methods depending on the number of mapped reads. The run time of the MCMC algorithm scales almost linearly with the number of reads. On the other hand, the run time of the VB algorithm seems to grow more than just by a linear increase.

4.2.5 Using approximate posteriors in differential expression analysis

We have shown that the variational method performs well in estimating the mean of the transcript expression, but underestimates the variance for a substantial fraction of transcripts. Here we investigate the effects of this underestimation on

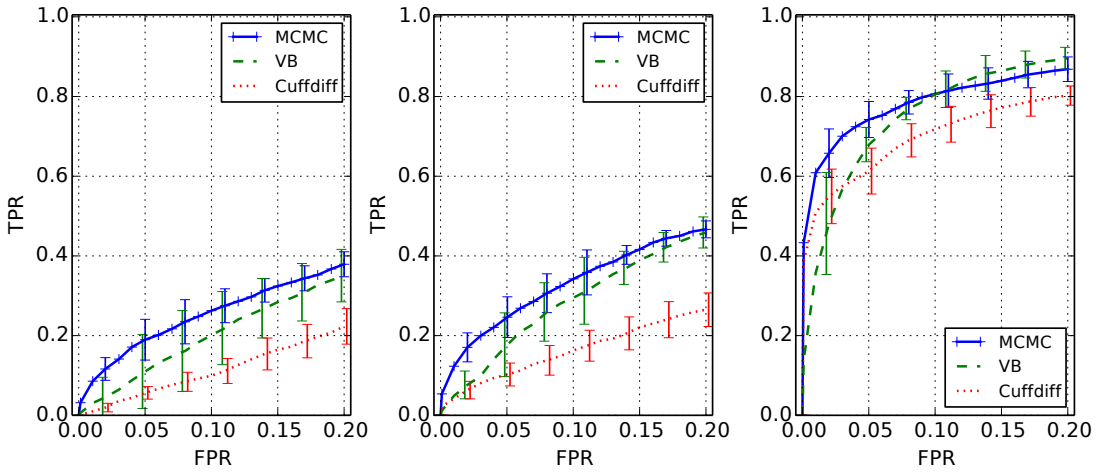


Figure 4.7: **Comparison of ROC curves for DE analysis of synthetic data using BitSeq with MCMC posterior, BitSeq with approximate posterior (VB) and Cufflinks.** Transcripts were split into three equal-sized groups based on the average true read count: low expression transcripts $[1, 3.75)$, intermediate expression transcripts $[3.75, 26.38)$ and high expression transcripts above 26.38. The ROCs are averaged over 5 independent analyses with different transcripts being differentially expressed, with two standard deviations as error bars. Using BitSeq with MCMC inference yields better and more stable performance.

a differential expression analysis between two replicated conditions.

In order to compare to a ground truth, we return to the synthetic data consisting of two conditions with two replicates each, which was previously introduced in Chapter 3. Expression of one third of transcripts was changed in one of the conditions, with fold change being uniformly selected from interval $[1.5, 3.5]$.

We use expression estimates obtained by MCMC and VB inference methods in combination with BitSeq differential expression analysis procedure. For comparison, we also compare against alternative approach using Cuffdiff (Trapnell et al., 2013). Figure 4.7 shows ROC characteristics of the different approaches for transcripts grouped into three groups based on initial mean expression. We can see that using MCMC expression estimates on average outperforms the use of VB estimates in terms of True Positive Rate.

BitSeq differential expression analysis estimates the Probability of Positive Log Ratio (PPLR) for each transcript. PPLR close to 1 signifies high probability of up-regulation, whereas values close to 0 mean high probability of down-regulation. The PPLR is then used for ranking transcripts in terms of differential

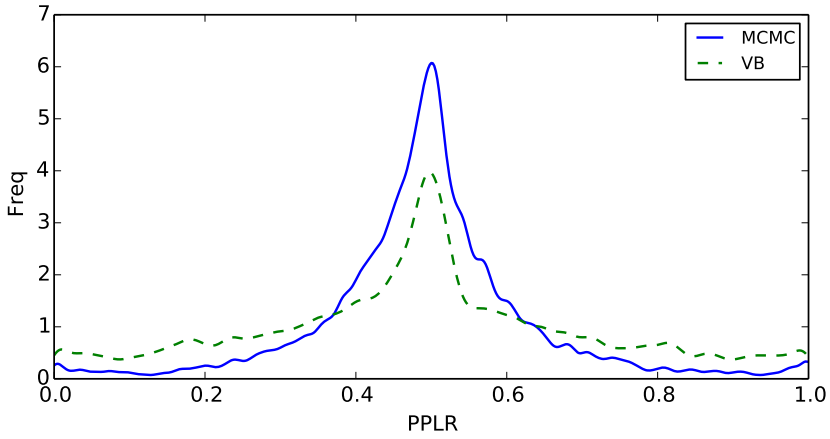


Figure 4.8: **Kernel density estimate of transcripts' Probability of Positive Log Ratio obtained by BitSeq differential expression analysis.** The PPLR was estimated using BitSeq DE analysis based on expression estimates obtained by MCMC and VB inference algorithms. The distribution of transcripts' PPLR was smoothed using Gaussian kernel density estimator. Using expression estimates from VB inference method results in more extreme values of transcripts' PPLR.

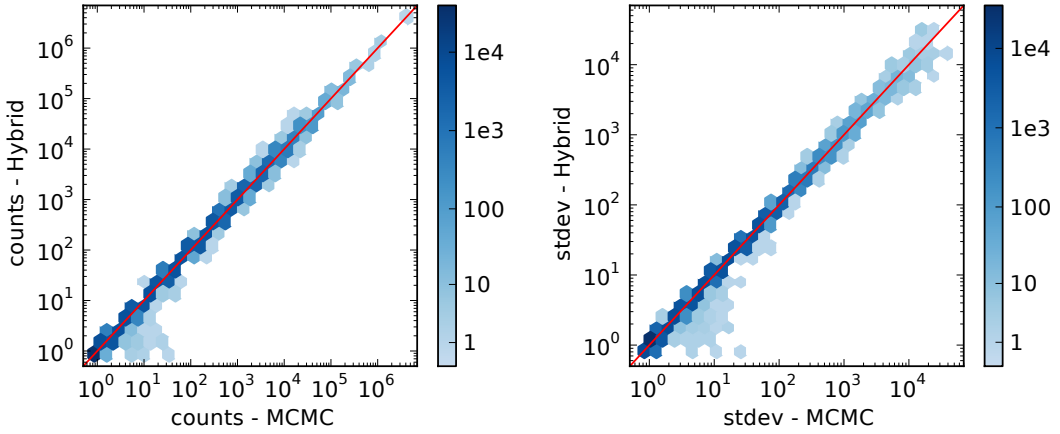
expression likelihood and selecting significant differences. In Figure 4.8 we show smoothed distribution of transcripts' PPLR produced by BitSeq when used with either MCMC or VB expression estimates. Due to the underestimation of variance in the VB inference approach, the resulting PPLR tends to more extreme values in terms of differential expression likelihood.

4.2.6 Combining Variational Bayes with Gibbs sampling

Previous results show that while Variational Bayes inference underestimates the variance, it can provide reliable estimates of mean expression levels much faster than MCMC algorithm. We can use this property of VB to improve efficiency of MCMC by combining these two algorithms.

First of all, the Markov chains are in our case initialised randomly from the prior distribution followed by burn-in phase. The purpose of burn-in phase is for the chains to converge to the posterior distribution, thus all samples from burn-in are discarded. Initialising chains from the posterior distribution inferred by VB algorithm enables us to shorten the burn-in.

The second improvement is based on the idea that while assignment of some reads is ambiguous, many reads can be easily assigned to a specific transcript.



(a) Posterior mean transcript expression

(b) Posterior standard deviation in transcript expression

Figure 4.9: **A comparison of the first two moments of the posterior distribution inferred by the hybrid VB-MCMC algorithm.** The posterior distributions of transcript expression levels in counts sampled by the hybrid VB-MCMC method and a long run of MCMC are compared in terms of: (a) posterior mean (R^2 correlation is 0.999) (b) posterior standard deviation.

The VB algorithm optimises the approximate distribution through the variable ϕ, where ϕ_{nm} expresses the posterior probability of $z_n m = 1$, e.g. the probability that the read r_n originates from transcript m. Reads that have ϕ_{nm} close to 1 for certain transcript m can be assigned to that transcript and do not have to be re-sampled in every iteration of Gibbs sampling.

Here we applied the second approach mentioned above to improve computational efficiency of MCMC algorithm. We analysed the dataset used in Section 4.2.2 which contains 98.8M paired-end reads and was generated by ENCODE consortium. After running the VB inference, we used the posterior distribution to initialise chains and assigned reads with high assignment likelihoods to a fixed transcript based on ϕ. We used 0.999 as the threshold over ϕ_{nm} to select reads that were assigned to a single transcript, resulting in 35.3M fixed reads out of 98.8M total. We followed the VB with collapsed Gibbs sampling applied to the rest of the reads combined with counts of the fixed reads.

Figure 4.9 shows the first two moments of the inferred posterior distribution compared with distribution obtained by a long run of MCMC. Except for the few outliers, the hybrid VB-MCMC algorithm produces posterior distribution with correctly estimated mean and variance.

The run time of the method without the pre-processing stage was 1391 minutes, compared with the 1607 minutes required by the MCMC algorithm (see Table 4.2). While this saves 3.6 hours out of 26.8 hours for the MCMC algorithm, the speed-up is not as significant as we would hope to achieve. Initialising MCMC with VB estimate and shortening the burn-in period could potentially save extra 2-3 hours of run time on this relatively big dataset. Furthermore the high memory usage (around 26GB) is still required due to VB optimisation.

4.3 Summary and related work

We have presented a variational method for transcript expression level inference from RNA-seq data. Building on previous work using MCMC and advances in VB with conjugate gradient optimisation, we have presented a fast approximate inference method. We have shown that the mean of the approximate posterior inferred by VB provides equivalent level of accuracy as mean inferred by MCMC algorithm.

The VB inference provides an efficient alternative to the MCMC approach with almost 10 fold speed up on large RNA-seq dataset, with realistic memory requirements. For applications in which only the point estimate of expression is required, VB algorithm is a major improvement.

Our VB method can also benefit from parallel processing, in that one of the expensive computations – taking logarithms and exponents to convert between ϕ and γ can be parallelised. Preliminary runs show good speed-up for this method on a multiple-core machine, where this loop can be tightly parallelised.

We have compared our method with recently published VB inference in the TIGAR approach (Nariai et al., 2013). We conclude that while TIGAR showed slightly improved accuracy in our comparison, the TIGAR implementation requires large amounts of memory, and does not offer a significant improvement over Gibbs sampling in terms of time efficiency. Furthermore, TIGAR at the moment does not provide means of accounting for read distribution bias and assumes uniform read distribution. While this is not a problem on simulated data with reads sampled uniformly, for real datasets accounting for these biases has been shown to improve accuracy (Roberts et al., 2011).

Our experimental results showed that the VB inference leads to underestimation of posterior variance. This can be explained by the independence assumption

in the approximate distribution $q(\boldsymbol{\theta}, \mathbf{Z})$. While this does not affect accuracy of our method in terms of mean expression, methods that use whole posterior distribution will be affected. Comparisons of DE analysis based on VB posterior showed that the variational approximation does not work as well as Gibbs sampling, though it does offer some improvement over the Cuffdiff method.

We explored the alternative of using VB optimisation to improve the efficiency of MCMC algorithm. Our preliminary analysis showed good results in terms of the mean and variance accuracy of the posterior distribution of transcript expression levels. However the relatively modest improvements in run time efficiency do not provide sufficient justification for this approximation at the moment.

Related work

Our major concern of the current VB inference is the underestimation of the posterior variance. A recent work by Papastamoulis et al. (2013) further extends the VB inference method presented here. In the report, the inference method avoids the independence assumption in $q(\boldsymbol{\theta}, \mathbf{Z})$ and applies stochastic approximation algorithm to infer the posterior distribution. This leads to improved assessment of uncertainty of the final estimate in form of more realistic posterior variance. The new approach further uses the posterior in form of a Generalized Dirichlet distribution which can capture the correlation between variables observed in the empirical posterior of MCMC (see Section 2.4.2).

Furthermore, the report proposes improved bound on the marginal likelihood used in the VB optimisation. Except for its implications for the inference algorithm, marginal likelihood can be used for model selection. This can be exploited in future applications relying on marginal likelihood for selecting between different bias models or choosing a correct set of splice variants.

Chapter 5

Conclusion

5.1 Accomplished results

In this thesis we have presented novel probabilistic approaches for transcript and gene expression quantification and differential expression analysis. We have shown that they provide accurate results, comparable with alternative methods with the advantage of a certainty measure in form of a full posterior distribution over the inferred parameters.

The research project presented within this thesis spans at least three major scientific fields. Biology, for understanding genetics and principles of the high-throughput sequencing. Statistics, used in probabilistic modelling and Bayesian inference. And lastly, Computer science, providing means for effective implementations of the novel methods and principles of software engineering. Here we provide an overview of the initial objectives and evaluate their fulfillment.

Method for transcript expression quantification based on RNA-seq data, providing accurate results with estimate of uncertainty

We have developed a new method for transcript expression level estimation from high-throughput sequencing data. The method uses a probabilistic generative model of the observed reads and extends previously published models by Li et al. (2010a) and Nicolae et al. (2011). The model accounts for read errors, paired-end read fragment lengths and non-uniform read distributions.

Given empirically estimated likelihoods of alignments, the method uses full Bayesian inference procedure. While the exact posterior distribution of expression levels is intractable, due to the use of a conjugate model, we can apply the Gibbs

sampling algorithm. The convergence of the sampling algorithm is monitored by using multiple parallel chains and comparing the within chain and between chain variances. Samples from the posterior distribution of transcript expression can be transformed into alternative measure of expression such as RPKM, or combined into gene expression. Thanks to generating full posterior distribution of expression estimates, a level of certainty of the estimates can be assessed either from the full distribution or by its variance and percentiles. The full posterior additionally enables detection of correlations between highly similar transcripts.

Our evaluations on real RNA-seq data show good correlation with qRT-PCR validation, which can be further improved by using the non-uniform read distribution bias model. Using synthetic data, we showed that our method provides highest level of accuracy when estimating the within-gene proportions of transcripts and provides results on a par with alternative approaches in terms of absolute transcript and gene expression levels.

Efficient inference algorithms for RNA-seq data analysis and transcript expression quantification

Considering data with thousands of transcripts and hundred millions of read alignments, the use of a sampling algorithm can lead to unusable implementations. First of all, we have improved the standard Gibbs sampling algorithm by using the collapsed model. The collapsed Gibbs sampler, while having comparable speed per iteration, uses around half of the iterations of the standard Gibbs sampler to reach the same level of convergence in terms of the marginal posterior variance estimate. This can half the run time of the algorithm for a given convergence level.

Secondly, through optimised implementation and use of parallelisation we have created efficient application that can process and analyse data with almost hundred million mapped read pairs within 30 hours. Given that sequencing of the data takes several days, we consider this to be within acceptable time frame for the analysis.

Lastly, we have developed an alternative inference technique for the transcript expression estimation problem in form of a Variational Bayes algorithm. The Variational Bayes inference is based on approximation of the posterior distribution with other, tractable, distribution and minimising the divergence between the posterior and approximation. It provides a faster alternative for estimating

the mean expression, with up to 10 fold speed up in comparison with the collapsed Gibbs samples. However, the estimate is penalised by underestimation of the true variance of the posterior distribution.

Probabilistic method for transcript-level differential expression analysis accounting for biological variance and using expression with uncertainty measure

Building on the results of our transcript expression quantification approach we have created a new method for differential expression analysis that uses input in form of probability distribution of expression levels. At the time being, all other methods either take as an input read alignments or use a point estimate of expression. Furthermore, most commonly used methods rely on read counts in combination with either Poisson or Negative-Binomial model. While these can be used for transcript expression, we argued that read counts should not be used for gene level DE analysis because of varying effective lengths of genes. Our approach uses Log-Normal model with the input of expression samples in the RPKM measure and can be applied to both transcript and gene level analysis.

Our method relies on the use of biological replicates for estimation of transcript-specific abundance fluctuations within condition. The use of biological replication and assessment of biological variance is important for determining the true condition dependent changes. This has been the norm for microarray DE analysis and applies the same way to RNA-seq.

We have evaluated our method on artificial dataset and provided a comparison with alternative DE methods. We have shown improved performance in DE detection of transcripts. However, we acknowledge the need for a better way of validating DE analysis approaches, either through improved simulation of data or through extensively validated real RNA-seq datasets.

Usable implementation available to other researchers

All of the methods in this thesis have been implemented with intent of providing useful application for other researchers. The tools are part of the BitSeq package which is freely distributed under permissive open-source license, *Artistic License 2.0*, with exception of external libraries that are distributed under various other open-source licenses. While most of the analysis can be performed on a standard

desktop computer, we advise the use of a dedicated computing node for the analysis of larger datasets.

The package is mainly implemented in `C++` with two alternative versions. The first version is distributed as a standalone compilable source code for GNU Linux and Unix based operating systems. This version provides individual command line programs for inference and data manipulation. The second version is part of the Bioconductor project and provides interface between the command line binaries and the `R` environment for statistical computing.

5.2 Research output

The work presented within this thesis led to several scientific publications, a software implementation and conference presentations.

Publications:

- P. Glaus, A. Honkela, and M. Rattray. Identifying differentially expressed transcripts from RNA-seq data with biological variation. *Bioinformatics*, 28(13):1721–8, July 2012.
- J. Hensman, P. Glaus, A. Honkela, and M. Rattray. Fast Approximate Inference of Transcript Expression Levels from RNA-seq Data. (In preparation), 2013.
- P. Papastamoulis, J. Hensman, P. Glaus, and M. Rattray. Improved Variational Bayes inference for transcript expression estimation. *Statistical Applications in Genetics and Molecular Biology*, (Accepted), 2013.

Software package *BitSeq*, available as:

- `C++`, command line: `http://code.google.com/p/bitseq/`
- `R`, Bioconductor: `http://bioconductor.org/packages/release/bioc/html/BitSeq.html`

Selected presentations:

21.7.2013 Peter Glaus. Identifying differentially expressed transcripts from RNA-seq data with biological variation. *ISMB/ECCB, Berlin*, 2013.

1.6.2012 Peter Glaus. Bayesian Inference of Transcripts from Sequencing data. *SeqAhead workshop, Uppsala*, 2012.

15.7.2011 Peter Glaus. Estimating differential expression of transcripts with RNA-seq by using Bayesian inference. *HitSeq, Vienna*, 2011.

5.3 Future work

We have presented completed methods for transcript expression quantification and DE analysis which can be applied to real life analysis of RNA-seq datasets. However, thanks to the broad range of problems associated with these tasks, there are multiple paths for future research and development of this project.

Transcript expression quantification

One of the important future research directions of the transcript expression quantification problem is the improvement of Variational Bayes inference techniques. We have already mentioned current work by Papastamoulis et al. (2013), which improves the Variational inference. Papastamoulis et al. (2013) propose various modifications to the current VB inference through the use of approximate distributions without the assumption of independence between expression and read assignments; providing a better bound for marginal likelihood; and using Generalized Dirichlet distribution to approximate the posterior. These advancements can lead to faster inference method than the collapsed Gibbs sampling, without the loss in form of underestimated variance.

Moreover, the Variational Bayes approximations are also interesting because they provide a bound on the marginal likelihood. As we have noted earlier, in the Bayesian framework the marginal likelihood can be used for model selection, enabling comparison of different priors (see Nariai et al., 2013), or choosing between different read bias distribution models. Its most interesting application is the comparison of different splicing annotations. The estimated marginal likelihood can be used to choose a splicing model that describes the data the best.

Methods presented here were specifically designed for data generated by the *Second generation sequencing* technologies. With the rise of single cell sequencing (Brennecke et al., 2013) and the third generation of sequencing technologies (Branton et al., 2008; Schadt et al., 2010), new data will require different kind

of analysis. For example, the single-molecule real-time sequencing is capable of sequencing reads up to length of $10kb$, however with a much lower base accuracy (Shin et al., 2013). While long reads can resolve the issue of similar transcripts, probabilistic models similar to those proposed here can be used to account for the base errors within reads.

Differential expression analysis

As we have discussed in Chapter 3, False Discovery Rate is often used to select significance threshold when it comes to DE calls. While we have investigated use of the pFDR measure, it tends towards underestimation of the true number of false positive calls. More careful investigation of the pFDR estimation method might yield better calibrated results.

Our differential expression model can be applied to both transcript and gene RPKM expression. However, the model is not applicable to the within-gene relative expression of transcripts, due to different range of values and variance dependence. For detecting changes of splicing patterns within genes a new method has to be developed which will account for the relative expression and variance affected by the abundance of the particular transcript, its gene and other splice variants.

There are also several directions of extending the existing DE model itself. One possibility lies in integrating the quantification step with the differential expression analysis. As this would involve joint expression estimation of multiple experiments, using sampling algorithms would be computationally impossible. However, the Variational methods provide efficient alternative and could be used for inference instead of Gibbs sampling.

Another alternative involves extension of the model in order to enable multi-factor analysis that involves comparison of multiple conditions that can be grouped into factors. As an example, it could be a comparison of normal condition and treatment, both at two different time points. Our model does allow joint DE analysis of all conditions and time points, however it only provides pairwise comparison of samples. Additional level of hierarchy in the model could allow more complex comparisons such as changes between time points irrespective of condition.

Inference and implementation

In chapter 4 we have presented a hybrid inference method, which combined the Variational Bayes inference algorithm with Gibbs sampling in order to improve the run time of the sampling algorithm. The method uses VB to estimate initial expression with subsequent stable assignment of subset of reads that have high posterior likelihood of assignment to a single transcript. We showed that the method provides more accurate estimate of variance as well shorter run time than Gibbs sampling. However, we would like to improve the speed-up of the algorithm through finding better ways of combining the two inference methods.

An alternative way of improving the speed of the Gibbs sampler could be through the application of similar technique as was used for speeding up the Gibbs sampling in the Latent Dirichlet Allocation (LDA) (Porteous et al., 2008). LDA is a method used for assigning documents to unobserved topics using generative model (Blei et al., 2003). The probabilistic model used in our expression estimation approach can be viewed as a subset of the model used for LDA. Porteous et al. (2008) showed an alternative assignment process within the collapsed Gibbs sampler that avoids calculation of all assignment likelihoods and provides dramatic speed-up for the LDA. It would be interesting to investigate whether a similar technique could be used for speeding up the collapsed Gibbs sampler used in our application.

We have developed an application for quantifying transcript expression and DE analysis, which is intended for use by other researchers. A certain portion of our future efforts has to be aimed at maintaining the application. This involves improving the usability of the application and providing documentation and support for other researchers using the application. Similarly, it is also important to accommodate future trends in RNA-seq, sequencing datasets and use cases of the high-throughput sequencing technology.

Bibliography

M. I. Abouelhoda, S. Kurtz, and O. Enno. The Enhanced Suffix Array and Its Applications to Genome Analysis. In *Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, WABI '02, pages 449–463. Springer-Verlag, 2002.

C. Alkan, S. Sajjadian, and E. E. Eichler. Limitations of next-generation genome sequence assembly. *Genome Biology*, 8(1):61–65, 2011.

S. I. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

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

S. Anders, A. Reyes, and W. Huber. Detecting differential usage of exons from RNA-seq data. *Genome Research*, 22(10):2008–17, October 2012.

K. F. Au, H. Jiang, L. Lin, Y. Xing, and W. H. Wong. Detection of splice junctions from paired-end RNA-seq data by SpliceMap. *Nucleic Acids Research*, 38(14): 4570–8, August 2010.

P. L. Auer and R. W. Doerge. Statistical Design and Analysis of RNA-Seq Data. *Genetics*, May 2010.

P. L. Auer and R. W. Doerge. A Two-Stage Poisson Model for Testing RNA-Seq Data. *Statistical Applications in Genetics and Molecular Biology*, 10(1):1–26, January 2011.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

C. M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning*, 3(4-5):993–1022, 2003.

R. Bohnert and G. Rätsch. rQuant.web: a tool for RNA-Seq-based transcript quantitation. *Nucleic Acids Research*, 38(Web Server issue):W348–51, July 2010.

R. Bohnert, J. Behr, and G. Rätsch. Transcript quantification with RNA-Seq data. *BMC Bioinformatics*, 10(Suppl 13):P5, 2009.

D. Branton, D. W. Deamer, A. Marziali, H. Bayley, S. A. Benner, T. Butler *et al.* The potential and challenges of nanopore sequencing. *Nature Biotechnology*, 26(10):1146–53, October 2008.

P. Brennecke, S. Anders, J. K. Kim, A. a. Koodziejczyk, X. Zhang, V. Proserpio *et al.* Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, (September), September 2013.

J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC bioinformatics*, 11:94, January 2010.

M. Burrows and D. J. Wheeler. A Block-Sorting Lossless Data Compression Algorithm. 1994.

Y. Chen, T. Souaiaia, and T. Chen. PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds. *Bioinformatics*, 25(19):2514–21, 2009.

W. S. Cleveland. Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, 74(368):829, December 1979.

M. David, M. Dzamba, D. Lister, L. Ilie, and M. Brudno. SHRiMP2: Sensitive yet Practical Short Read Mapping. *Bioinformatics*, 27(7):1011–1012, January 2011.

F. De Bona, S. Ossowski, K. Schneeberger, and G. Rätsch. Optimal spliced alignments of short sequence reads. *Bioinformatics*, 24(16):i174–80, August 2008.

M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, 43(5):491–8, May 2011.

S. Djebali, C. A. Davis, A. Merkel, A. Dobin, T. Lassmann, A. Mortazavi *et al.* Landscape of transcription in human cells. *Nature*, 489(7414):101–8, September 2012.

A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, January 2013.

J. C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, 36(16):e105, September 2008.

P. Drewe, O. Stegle, L. Hartmann, A. Kahles, R. Bohnert, A. Wachter *et al.* Accurate detection of differential RNA processing. *Nucleic Acids Research*, 41 (10):5189–98, May 2013.

S. Dudoit, Y. Yang, M. Callow, and T. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica sinica*, 12:111–139, 2002.

Z. Fang and X. Cui. Design and validation issues in RNA-seq experiments. *Briefings in bioinformatics*, 12(3):280–7, May 2011.

P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, page 390. IEEE Computer Society, November 2000.

R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, February 1964.

P. Flicek, I. Ahmed, M. R. Amode, D. Barrell, K. Beal, S. Brent *et al.* Ensembl 2013. *Nucleic Acids Research*, 41(Database issue):D48–55, January 2013.

N. A. Fonseca, J. Rung, A. Brazma, and J. C. Marioni. Tools for mapping high-throughput sequencing data. *Bioinformatics*, 28(24):3169–77, December 2012.

M. Garber, M. G. Grabherr, M. Guttman, and C. Trapnell. Computational methods for transcriptome annotation and quantification using RNA-seq. *Nature Methods*, 8(6):469–77, June 2011.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, second edition, 2004.

S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit *et al.* Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, January 2004.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, London, 1 edition, December 1995.

M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. a. Thompson, I. Amit *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology*, 29(7):644–52, July 2011.

G. R. Grant, M. H. Farkas, A. D. Pizarro, N. F. Lahens, J. Schug, B. P. Brunk *et al.* Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, 27(18):2518–28, September 2011.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228–35, April 2004.

M. Guttman, M. Garber, J. Z. Levin, J. Donaghey, J. Robinson, X. Adiconis *et al.* Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nature Biotechnology*, 28 (5):503–10, May 2010.

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, July 2010.

K. D. Hansen, R. A. Irizarry, and Z. Wu. Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics*, 13(2):204–16, April 2012.

T. J. Hardcastle and K. A. Kelly. baySeq: Empirical Bayesian Methods For Identifying Differential Expression In Sequence Count Data. *BMC Bioinformatics*, 11(1):422, 2010.

J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski *et al.* GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Research*, 22(9):1760–74, September 2012.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

A.-M. K. Hein and S. Richardson. A powerful method for detecting differentially expressed genes from GeneChip arrays that does not require replicates. *BMC bioinformatics*, 7(1):353, January 2006.

J. Hensman, M. Rattray, and N. Lawrence. Fast Variational Inference in the Conjugate Exponential Family. In P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2897–2905. 2012.

M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49 (6):409–436, December 1952.

B. G. Hoffman and S. J. M. Jones. Genome-wide identification of DNA-protein interactions using chromatin immunoprecipitation coupled with flow cell sequencing. *The Journal of endocrinology*, 201(1):1–13, April 2009.

N. Homer, B. Merriman, and S. F. Nelson. BFAST: an alignment tool for large scale genome resequencing. *PloS one*, 4(11):e7767, January 2009.

A. Honkela, T. Raiko, M. Kuusela, M. Tornio, and J. Karhunen. Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes. *Journal of Machine Learning Research*, 11:3235–3268, 2010.

B. E. Howard and S. Heber. Towards reliable isoform quantification using RNA-SEQ data. *BMC bioinformatics*, 11 Suppl 3(Suppl 3):S6, January 2010.

F. Hsu, W. J. Kent, H. Clawson, R. M. Kuhn, M. Diekhans, and D. Haussler. The UCSC Known Genes. *Bioinformatics*, 22(9):1036–46, May 2006.

Human Genome Sequencing Consortium International. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–45, October 2004.

H. Jiang and W. H. Wong. Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics*, 25(8):1026–32, April 2009.

D. Karolchik, A. S. Hinrichs, T. S. Furey, K. M. Roskin, C. W. Sugnet, D. Haussler *et al.* The UCSC Table Browser data retrieval tool. *Nucleic Acids Research*, 32(Database issue):D493–6, January 2004.

Y. Katz, E. T. Wang, E. M. Airoldi, and C. B. Burge. Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nature Methods*, 7: 1009–1015, November 2010.

W. J. Kent. BLAT—The BLAST-Like Alignment Tool. *Genome Research*, 12 (4):656–664, March 2002.

D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biology*, 14(4):R36, April 2013.

H. Kim, Y. Bi, S. Pal, R. Gupta, and R. V. Davuluri. IsoformEx: isoform level gene expression estimation using weighted non-negative least squares from mRNA-Seq data. *BMC bioinformatics*, 12(1):305, January 2011.

R. M. Kuhn, D. Haussler, and W. J. Kent. The UCSC genome browser and associated tools. *Briefings in bioinformatics*, 14(2):144–61, March 2013.

V. M. Kvam, P. Liu, and Y. Si. A comparison of statistical methods for detecting differentially expressed genes from RNA-seq data. *American journal of botany*, 99(2):248–56, February 2012.

T. D. Laajala, S. Raghav, S. Tuomela, R. Lahesmaa, T. Aittokallio, and L. L. Elo. A practical comparison of methods for detecting transcription factor binding sites in ChIP-seq experiments. *BMC genomics*, 10:618, January 2009.

V. Lacroix, M. Sammeth, R. Guigo, and A. Exact transcriptome reconstruction from short sequence reads. In *Algorithms in Bioinformatics*, volume 5251/2008, pages 50–63. Springer Berlin / Heidelberg, 2008.

B. Langmead and S. L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–9, April 2012.

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

S. Lee, C. H. Seo, B. Lim, J. O. Yang, J. Oh, M. Kim *et al.* Accurate quantification of transcriptome from RNA-Seq data by effective length normalization. *Nucleic Acids Research*, 39(2):e9, January 2011.

N. Leng, J. a. Dawson, J. a. Thomson, V. Ruotti, A. I. Rissman, B. M. G. Smits *et al.* EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics*, 29(8):1035–43, April 2013.

J. Z. Levin, M. Yassour, X. Adiconis, C. Nusbaum, D. A. Thompson, N. Friedman *et al.* Comprehensive comparative analysis of strand-specific RNA sequencing methods. *Nature Methods*, 7(9):709–715, August 2010.

B. Li and C. N. Dewey. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12(1):323, 2011.

B. Li, V. Ruotti, R. M. Stewart, J. A. Thomson, and C. N. Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26 (4):493–500, 2010a.

H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–60, 2009.

H. Li and N. Homer. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in bioinformatics*, 11(5), May 2010.

J. J. Li, C.-R. Jiang, J. B. Brown, H. Huang, and P. J. Bickel. Sparse linear modeling of next-generation mRNA sequencing (RNA-Seq) data for isoform discovery and abundance estimation. *Proceedings of the National Academy of Sciences of the United States of America*, 108(50):19867–72, December 2011a.

J. Li, H. Jiang, and W. H. Wong. Modeling non-uniformity in short-read rates in RNA-Seq data. *Genome Biology*, 11(5):R50, January 2010b.

R. Li, Y. Li, K. Kristiansen, and J. Wang. SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–4, March 2008.

R. Li, Y. Li, X. Fang, H. Yang, J. Wang, K. Kristiansen *et al.* SNP detection for massively parallel whole-genome resequencing. *Genome Research*, 19(6): 1124–32, July 2009a.

R. Li, C. Yu, Y. Li, T.-W. Lam, S.-M. Yiu, K. Kristiansen *et al.* SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966–7, August 2009b.

R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi *et al.* De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20 (2):265–72, February 2010c.

W. Li, J. Feng, and T. Jiang. IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. *Journal of computational biology : a journal of computational molecular cell biology*, 18(11):1693–707, November 2011b.

R. Lindner and C. C. Friedel. A comprehensive evaluation of alignment algorithms in the context of RNA-seq. *PloS one*, 7(12):e52403, January 2012.

J. S. Liu. The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem. *Journal of the American Statistical Association*, 89(427):958, September 1994.

L. Liu, Y. Li, S. Li, N. Hu, Y. He, R. Pong *et al.* Comparison of next-generation sequencing systems. *Journal of biomedicine & biotechnology*, 2012:251364, January 2012.

X. Liu, M. Milo, N. D. Lawrence, and M. Rattray. Probe-level measurement error improves accuracy in detecting differential gene expression. *Bioinformatics*, 22 (17):2107–13, September 2006.

G. Lunter and M. Goodson. Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Research*, 21(6):936–9, June 2011.

C. A. Maher, C. Kumar-Sinha, X. Cao, S. Kalyana-Sundaram, B. Han, X. Jing *et al.* Transcriptome sequencing to detect gene fusions in cancer. *Nature*, 458 (7234):97–101, March 2009.

E. R. Mardis. Next-generation DNA sequencing methods. *Annual review of genomics and human genetics*, 9:387–402, January 2008.

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):1509–17, 2008.

J. A. Martin and Z. Wang. Next-generation transcriptome assembly. *Nature Reviews. Genetics*, 12(10):671–82, October 2011.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087, 1953.

M. L. Metzker. Sequencing technologies - the next generation. *Nature Reviews. Genetics*, 11(1):31–46, January 2010.

L. R. Meyer, A. S. Zweig, A. S. Hinrichs, D. Karolchik, R. M. Kuhn, M. Wong *et al.* The UCSC Genome Browser database: extensions and updates 2013. *Nucleic Acids Research*, 41(Database issue):D64–9, January 2013.

J. R. Miller, S. Koren, and G. Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–27, June 2010.

R. D. Morin, M. Bainbridge, A. Fejes, M. Hirst, M. Krzywinski, T. J. Pugh *et al.* Profiling the HeLa S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *BioTechniques*, 45(1):81–94, 2008.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5 (7):621–8, 2008.

U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein *et al.* The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science (New York, N.Y.)*, 320(5881):1344–9, June 2008.

N. Nariai, O. Hirose, K. Kojima, and M. Nagasaki. TIGAR: transcript isoform abundance estimation method with gapped alignment of RNA-Seq data by variational Bayesian inference. *Bioinformatics*, 29(18):2292–2299, July 2013.

NCBI. The NCBI Sequence Read Archive, 2010. URL `http://www.ncbi.nlm.nih.gov/sra`.

M. Nicolae, S. Mangul, I. I. Mndoiu, and A. Zelikovsky. Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for molecular biology : AMB*, 6(1):9, January 2011.

R. Nielsen, J. S. Paul, A. Albrechtsen, and Y. S. Song. Genotype and SNP calling from next-generation sequencing data. *Nature Reviews. Genetics*, 12 (6):443–51, June 2011.

A. Oshlack, M. D. Robinson, and M. D. Young. From RNA-seq reads to differential expression results. *Genome Biology*, 11(12):220, December 2010.

P. Papastamoulis, J. Hensman, P. Glaus, and M. Rattray. Improved Variational Bayes inference for transcript expression estimation. *Statistical Applications in Genetics and Molecular Biology*, (Accepted), 2013.

P. J. Park. ChIP-seq: advantages and challenges of a maturing technology. *Nature Reviews. Genetics*, 10(10):669–80, October 2009.

S. Pepke, B. Wold, and A. Mortazavi. Computation for ChIP-seq and RNA-seq studies. *Nature Methods*, 6(11 Suppl):S22–32, November 2009.

I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08*, page 569, 2008.

M. A. Quail, M. Smith, P. Coupland, T. D. Otto, S. R. Harris, T. R. Connor *et al.* A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC genomics*, 13(1):341, January 2012.

H. Richard, M. H. Schulz, M. Sultan, A. Nürnberger, S. Schrinner, D. Balzereit *et al.* Prediction of alternative isoforms from exon expression levels in RNA-Seq experiments. *Nucleic Acids Research*, 38(10):e112, June 2010.

G. Rizk and D. Lavenier. GASSST: Global Alignment Short Sequence Search Tool. *Bioinformatics*, 26(20):2534–2540, August 2010.

A. Roberts and L. Pachter. Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature Methods*, 10(1):71–3, January 2013.

A. Roberts, C. Trapnell, J. Donaghey, J. L. Rinn, and L. Pachter. Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biology*, 12(3):R22, January 2011.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.

G. Robertson, J. Schein, R. Chiu, R. Corbett, M. Field, S. D. Jackman *et al.* De novo assembly and analysis of RNA-seq data. *Nature Methods*, 7(11):909–12, November 2010.

M. D. Robinson and A. Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25, January 2010.

M. D. Robinson and G. K. Smyth. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, 9(2):321–32, April 2008.

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):139–40, January 2010.

M. Ruffalo, T. LaFramboise, and M. Koyutürk. Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics*, 27(20): 2790–6, October 2011.

S. L. Salzberg, A. M. Phillippy, A. Zimin, D. Puiu, T. Magoc, S. Koren *et al.* GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research*, 22(3):557–67, March 2012.

E. E. Schadt, S. Turner, and A. Kasarskis. A Window into Third Generation Sequencing. *Human molecular genetics*, 19(2):227–240, September 2010.

F. J. Sedlazeck, P. Rescheneder, and A. von Haeseler. NextGenMap: fast and accurate read mapping in highly polymorphic genomes. *Bioinformatics*, 29(21): 2790–1, November 2013.

S. Shen, J. W. Park, J. Huang, K. a. Dittmar, Z.-X. Lu, Q. Zhou *et al.* MATS: a Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data. *Nucleic Acids Research*, 40(8):1–13, February 2012.

J. Shendure and H. Ji. Next-generation DNA sequencing. *Nature Biotechnology*, 26(10):1135–45, October 2008.

C. Sherlock and G. Roberts. Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets. *Bernoulli*, 15(3):774–798, August 2009.

S. T. Sherry. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311, January 2001.

L. Shi, L. H. Reid, W. D. Jones, R. Shippy, J. a. Warrington, S. C. Baker *et al.* The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology*, 24(9): 1151–61, September 2006.

S. C. Shin, D. H. Ahn, S. J. Kim, H. Lee, T.-J. Oh, J. E. Lee *et al.* Advantages of Single-Molecule Real-Time Sequencing in High-GC Content Genomes. *PloS one*, 8(7):e68824, January 2013.

J. T. Simpson and R. Durbin. Efficient de novo assembly of large genomes using compressed data structures. *Genome Research*, 22(3):549–56, March 2012.

J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and I. Birol. ABySS: a parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–23, June 2009.

D. Singh, C. F. Orellana, Y. Hu, C. D. Jones, Y. Liu, D. Y. Chiang *et al.* FDM: a graph-based statistical method to detect differential transcription using RNA-seq data. *Bioinformatics*, 27(19):2633–40, October 2011.

D. A. Skelly, M. Johansson, J. Madeoy, J. Wakefield, and J. M. Akey. A powerful and flexible statistical framework for testing hypotheses of allele-specific gene

expression from RNA-seq data. *Genome Research*, 21(10):1728–37, October 2011.

J. D. Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):479–498, August 2002.

J. D. Storey. The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics*, 31(6):2013–2035, 2003.

D. J. Sugarbaker, W. G. Richards, G. J. Gordon, L. Dong, A. De Rienzo, G. Maulik *et al.* Transcriptome sequencing of malignant pleural mesothelioma tumors. *Proceedings of the National Academy of Sciences of the United States of America*, 105(9):3521–6, March 2008.

R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):pp. 267–288, 1996.

T. T. Torres, M. Metta, B. Ottenwälder, and C. Schlötterer. Gene expression profiling by massively parallel sequencing. *Genome Research*, 18(1):172–7, January 2008.

C. Trapnell, L. Pachter, and S. L. Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–11, 2009.

C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):516–520, May 2010.

C. Trapnell, D. G. Hendrickson, M. Sauvageau, L. Goff, J. L. Rinn, and L. Pachter. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature Biotechnology*, 31(1):46–53, January 2013.

E. Turro, S.-Y. Su, A. Gonçalves, L. J. M. Coin, S. Richardson, and A. Lewin. Haplotype and isoform specific expression estimation using multi-mapping RNA-seq reads. *Genome Biology*, 12(2):R13, January 2011.

P. K. Wall, J. Leebens-Mack, A. S. Chanderbali, A. Barakat, E. Wolcott, H. Liang *et al.* Comparison of next generation sequencing technologies for transcriptome characterization. *BMC genomics*, 10:347, January 2009.

K. Wang, D. Singh, Z. Zeng, S. J. Coleman, Y. Huang, G. L. Savich *et al.* MapSplice: accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*, 38(18):e178, October 2010a.

L. Wang, Z. Feng, X. Wang, X. Wang, and X. Zhang. DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics*, 26(1):136–138, 2010b.

X. Wang, Z. Wu, and X. Zhang. Isoform abundance inference provides a more accurate estimation of gene expression levels in RNA-seq. *Journal of Bioinformatics and Computational Biology*, 8(supp01):177–192, December 2010c.

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

A. P. M. Weber, K. L. Weber, K. Carr, C. Wilkerson, and J. B. Ohlrogge. Sampling the Arabidopsis transcriptome with massively parallel pyrosequencing. *Plant physiology*, 144(1):32–42, May 2007.

E. G. Wilbanks and M. T. Facciotti. Evaluation of algorithm performance in ChIP-seq peak detection. *PloS one*, 5(7):e11471, January 2010.

B. T. Wilhelm and J.-R. Landry. RNA-Seq-quantitative measurement of expression through massively parallel RNA-sequencing. *Methods (San Diego, Calif.)*, 48(3):249–57, 2009.

T. D. Wu and S. Nacu. Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, 26(7):873–81, April 2010.

Z. Wu, X. Wang, and X. Zhang. Using non-uniform read distribution models to improve isoform expression inference in RNA-Seq. *Bioinformatics*, 27(4): 502–8, February 2011.

G. Xu, C. Fewell, C. Taylor, N. Deng, D. Hedges, X. Wang *et al.* Transcriptome and targetome analysis in MIR155 expressing cells using RNA-seq. *RNA*, 16 (8):1610–22, August 2010.

D. R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18(5):821–9, May 2008.

S. Zheng and L. Chen. A hierarchical Bayesian model for comparing transcriptomes at the individual transcript isoform level. *Nucleic Acids Research*, 37 (10):e75, June 2009.

Appendix A

Derivations

A.1 Transcript expression model

We provide detailed derivations of the posterior distribution over the model parameters and the conditional distributions used in the Gibbs sampling algorithms. First we derive the posterior distribution over $\boldsymbol{I}, \boldsymbol{\theta}, \theta^{act}$ with marginalization of the noise indicator $\boldsymbol{Z^{act}}$, which is then used in both standard and collapsed Gibbs algorithms.

$$
\begin{aligned}
P(\boldsymbol{I}, \boldsymbol{\theta}, \theta^{act}|R) &= \sum_{\boldsymbol{Z^{act}}} P(\boldsymbol{I}, \boldsymbol{Z^{act}}, \boldsymbol{\theta}, \theta^{act}|R) \\
&\propto P(\boldsymbol{\theta})P(\theta^{act}) \sum_{\boldsymbol{Z^{act}}} \prod_{n=1}^{N} P(r_n|I_n)P(I_n|\boldsymbol{\theta}, Z_n^{act})P(Z_n^{act}|\theta^{act}) \\
&\propto P(\boldsymbol{\theta})P(\theta^{act}) \prod_{n=1}^{N} \sum_{Z_n^{act}=0,1} P(r_n|I_n)P(I_n|\boldsymbol{\theta}, Z_n^{act})P(Z_n^{act}|\theta^{act}) \\
&\propto P(\boldsymbol{\theta})P(\theta^{act}) \prod_{n=1}^{N} \Big(\delta(I_n > 0)P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta})\theta^{act} + \\
&\quad \delta(I_n = 0)P(r_n|noise)(1-\theta^{act}) \Big) \\
&\propto P(\boldsymbol{\theta})P(\theta^{act}) \prod_{n;I_n \neq 0} \left(P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta})\theta^{act} \right) \\
&\quad \prod_{n;I_n=0} \left(P(r_n|noise)(1-\theta^{act}) \right).
\end{aligned}
\tag{A.1}
$$

A.1.1 Standard Gibbs sampler

$$P(\alpha|\beta, R) = \frac{P(\alpha, \beta|R)}{P(\beta|R)} \propto P(\alpha, \beta|R)_\beta \tag{A.2}$$

We can use the rule A.2 applied to the posterior distribution given in Equation A.1 to derive the proportional form of the conditional distributions for each parameter. The conditional distributions are then used in the Gibbs sampler algorithm to sample the individual parameters in an iterative fashion.

For the read allocations $\boldsymbol{I}$, the allocations of individual reads are conditionally independent given $\boldsymbol{\theta}$ and θ^{act}, $P(\boldsymbol{I}|\boldsymbol{\theta}, \theta^{act}, R) = \prod_{n=1}^{N} P(I_n|\boldsymbol{\theta}, \theta^{act}, R)$, hence we can derive the conditional distribution for a single allocation:

$$\begin{aligned} P(I_n|\boldsymbol{\theta}, \theta^{act}, R) \propto & \left(P(\boldsymbol{\theta})P(\theta^{act}) \prod_{n;I_n \neq 0} \left(P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta})\theta^{act}\right) \right. \\ & \left. \prod_{n;I_n=0} \left(P(r_n|noise)(1-\theta^{act})\right) \right)_{\boldsymbol{\theta}, \theta^{act}, \boldsymbol{I}_{-n}} \\ \propto & \, \delta(I_n > 0)P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta})\theta^{act} + \\ & \, \delta(I_n = 0)P(r_n|noise)(1-\theta^{act}) \\ \propto & \, \delta(I_n > 0)P(r_n|I_n)\theta_{I_n}\theta^{act} + \delta(I_n = 0)P(r_n|noise)(1-\theta^{act}), \end{aligned} \tag{A.3}$$

where we use a shorthand notation for the Categorical distribution, $P(I_n = m|\boldsymbol{\theta}) = \theta_{I_n}$. The conditional distribution has the form of the Categorical distribution with parameters $\boldsymbol{\phi}$ defined below:

$$\begin{aligned} P(I_n|\boldsymbol{\theta}, \theta^{act}, R) &= \mathrm{Cat}(I_n|\boldsymbol{\phi_n}), \\ \phi_{n0} &= P(r_n|noise)(1-\theta^{act})/Z_n^{(\phi)}, \\ m \neq 0; \phi_{nm} &= P(r_n|m)\theta_m\theta^{act}/Z_n^{(\phi)}, \\ Z_n^{(\phi)} &= P(r_n|noise)(1-\theta^{act}) + \textstyle\sum_{m=1}^{M} P(r_n|m)\theta_m\theta^{act}. \end{aligned} \tag{A.4}$$

The conditional distributions for $\boldsymbol{\theta}$ and θ^{act} are derived in a similar fashion.

$$\begin{aligned} P(\boldsymbol{\theta}|\boldsymbol{I}, \theta^{act}, R) &\propto \Bigg(P(\boldsymbol{\theta}) P(\theta^{act}) \prod_{n;I_n \neq 0} \big(P(r_n|I_n) \mathrm{Cat}(I_n|\boldsymbol{\theta}) \theta^{act} \big) \\ &\qquad \prod_{n;I_n=0} \big(P(r_n|noise)(1-\theta^{act}) \big) \Bigg)_{\boldsymbol{I}, \theta^{act}} \\ &\propto P(\boldsymbol{\theta}) \prod_{n;I_n \neq 0} \mathrm{Cat}(I_n|\boldsymbol{\theta}) \\ &\propto \prod_{m=1}^{M} (\theta_m)^{\alpha^{dir}} \prod_{n;I_n \neq 0} \theta_{I_n} \\ &\propto \prod_{m=1}^{M} (\theta_m)^{\alpha^{dir}} \prod_{m=1}^{M} (\theta_m)^{C_m} = \prod_{m=1}^{M} (\theta_m)^{\alpha^{dir}+C_m}, \end{aligned} \tag{A.5}$$

where $C_m = \sum_{n=1}^{N} \delta(I_n = m)$ denotes the number of reads allocated to transcript m. Thanks to conjugacy of the Categorical-Dirichlet model, the conditional posterior distribution over $\boldsymbol{\theta}$ has the form of a Dirichlet distribution,

$$P(\boldsymbol{\theta}|\boldsymbol{I}, \theta^{act}, R) = \mathrm{Dir}(\boldsymbol{\theta}|\alpha^{dir} + C_1, \ldots, \alpha^{dir} + C_M). \tag{A.6}$$

Lastly, for θ^{act} we have

$$\begin{aligned} P(\theta^{act}|\boldsymbol{I}, \boldsymbol{\theta}, R) &\propto \Bigg(P(\boldsymbol{\theta}) P(\theta^{act}) \prod_{n;I_n \neq 0} \big(P(r_n|I_n) \mathrm{Cat}(I_n|\boldsymbol{\theta}) \theta^{act} \big) \\ &\qquad \prod_{n;I_n=0} \big(P(r_n|noise)(1-\theta^{act}) \big) \Bigg)_{\boldsymbol{I}, \boldsymbol{\theta}} \\ &\propto P(\theta^{act}) \prod_{n;I_n \neq 0} \theta^{act} \prod_{n;I_n=0} (1-\theta^{act}) \\ &\propto (\theta^{act})^{\alpha^{act}-1} (1-\theta^{act})^{\beta^{act}-1} (\theta^{act})^{\sum_{m=1}^{M} C_m} (1-\theta^{act})^{C_0} \\ &\propto (\theta^{act})^{\alpha^{act}+N-C_0-1} (1-\theta^{act})^{\beta^{act}+C_0-1}, \end{aligned} \tag{A.7}$$

which has the form of a Beta distribution with parameters $\alpha^{act} + N - C_0$ and $\beta^{act} + C_0$.

A.1.2 Collapsed Gibbs sampler

The collapsed Gibbs sampler is based on a collapsed model, with certain parameters marginalized out. In our case, in order to retain posterior in a usable form, we integrate over θ^{act} and $\boldsymbol{\theta}$ and sample the read allocations $\boldsymbol{I}$. We first integrate over θ^{act}:

$$
\begin{aligned}
P(\boldsymbol{I}, \boldsymbol{\theta}|R) &= \int_0^1 \mathrm{d}\theta^{act}\, P(\boldsymbol{I}, \boldsymbol{\theta}, \theta^{act}|R) \\
&\propto \int_0^1 \mathrm{d}\theta^{act}\, P(\boldsymbol{\theta}) P(\theta^{act}) \prod_{n;I_n\neq 0} \left(P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta})\theta^{act}\right) \\
&\quad \prod_{n;I_n=0} \left(P(r_n|noise)(1-\theta^{act})\right) \\
&\propto P(\boldsymbol{\theta}) \prod_{n;I_n\neq 0} P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta}) \prod_{n;I_n=0} P(r_n|noise) \\
&\quad \int_0^1 \mathrm{d}\theta^{act}\, P(\theta^{act})(\theta^{act})^{C_+}(1-\theta^{act})^{C_0} \\
&\propto P(\boldsymbol{\theta}) \prod_{n;I_n\neq 0} P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta}) \prod_{n;I_n=0} P(r_n|noise) \\
&\quad \int_0^1 \mathrm{d}\theta^{act}\, (\theta^{act})^{\alpha^{act}+C_+}(1-\theta^{act})^{\beta^{act}+C_0} \\
&\propto P(\boldsymbol{\theta}) \prod_{n;I_n\neq 0} P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta}) \prod_{n;I_n=0} P(r_n|noise) \\
&\quad \frac{\Gamma(\alpha^{act}+C_+)\Gamma(\beta^{act}+C_0)}{\Gamma(\alpha^{act}+\beta^{act}+N)},
\end{aligned}
\tag{A.8}
$$

where $C_+ = \sum_{m=1}^{M} C_m$ denotes the number of non-noise allocations. While the read allocations were conditionally independent given $\boldsymbol{\theta}$ and θ^{act}, after integrating out θ^{act} we loose the conditional independence. The posterior distribution does not factorize into independent factors for each read, because it contains the terms C_+ and C_0, which denote the total number of allocations to real and noise

transcripts respectively. We follow with marginalization of $\boldsymbol{\theta}$

$$
\begin{aligned}
P(\boldsymbol{I}|R) &= \int \mathrm{d}\boldsymbol{\theta}\, P(\boldsymbol{I},\boldsymbol{\theta}|R) \\
&\propto \Gamma(\alpha^{act}+C_+)\Gamma(\beta^{act}+C_0) \\
&\quad \int \mathrm{d}\boldsymbol{\theta}\, P(\boldsymbol{\theta}) \prod_{n;I_n\neq 0} P(r_n|I_n)\mathrm{Cat}(I_n|\boldsymbol{\theta}) \prod_{n;I_n=0} P(r_n|noise) \\
&\propto \Gamma(\alpha^{act}+C_+)\Gamma(\beta^{act}+C_0) \prod_{n;I_n\neq 0} P(r_n|I_n) \prod_{n;I_n=0} P(r_n|noise) \\
&\quad \int \mathrm{d}\boldsymbol{\theta} \prod_{m=1}^{M} (\theta_m)^{\alpha^{dir}-1} \prod_{n;I_n\neq 0} \theta_{I_n} \\
&\propto \Gamma(\alpha^{act}+C_+)\Gamma(\beta^{act}+C_0) \prod_{n;I_n\neq 0} P(r_n|I_n) \prod_{n;I_n=0} P(r_n|noise) \\
&\quad \int \mathrm{d}\boldsymbol{\theta} \prod_{m=1}^{M} (\theta_m)^{\alpha^{dir}+C_m-1} \\
&\propto \Gamma(\alpha^{act}+C_+)\Gamma(\beta^{act}+C_0) \prod_{n;I_n\neq 0} P(r_n|I_n) \prod_{n;I_n=0} P(r_n|noise) \\
&\quad \frac{\prod_{m=1}^{M}\Gamma(\alpha^{dir}+C_m)}{\Gamma(\sum_{m=1}^{M}\alpha^{dir}+C_m)}\,.
\end{aligned}
\tag{A.9}
$$

We cannot use this posterior distribution directly as it does not have a form of any standard probability distributions. However we apply the principle of Gibbs sampler and sample individual parameters, in this case allocations of single reads, conditioned on the other allocations. The conditional posterior distribution of single allocation is given below

$$
\begin{aligned}
P(I_n|I^{(-n)},R) &\propto \big(P(I_n,I^{(-n)}|R)\big)_{I^{(-n)}} \\
&\propto \delta(I_n=0)\Bigg(P(r_n|noise)\Gamma(\alpha^{act}+C_+^{(-n)})\Gamma(\beta^{act}+C_0^{(-n)}+1) \\
&\quad \frac{\prod_{m=1}^{M}\Gamma(\alpha^{dir}+C_m^{(-n)})}{\Gamma(M\alpha^{dir}+C_+^{(-n)})}\Bigg)+ \\
&\quad \delta(I_n>0)\Bigg(P(r_n|I_n)\Gamma(\alpha^{act}+C_+^{(-n)}+1)\Gamma(\beta^{act}+C_0^{(-n)}) \\
&\quad \frac{\Gamma(\alpha^{dir}+C_{I_n}^{(-n)}+1)\prod_{m>0;m\neq I_n}\Gamma(\alpha^{dir}+C_m^{(-n)})}{\Gamma(M\alpha^{dir}+C_+^{(-n)}+1)}\Bigg),
\end{aligned}
\tag{A.10}
$$

with $C_m^{(-n)} = \sum_{i \neq n} \delta(I_i = m)$ denoting the counts based on the other allocations and $C_+^{(-n)}$ defined accordingly. We now use the 'factorial' property of the Gamma function, $\Gamma(x+1) = x\Gamma(x)$, to simplify the conditional distribution:

$$\begin{aligned} P(I_n|I^{(-n)}, R) \propto \delta(I_n = 0)\Bigg(& P(r_n|noise)\Gamma(\alpha^{act} + C_+^{(-n)})\Gamma(\beta^{act} + C_0^{(-n)}) \\ & (\beta^{act} + C_0^{(-n)})\frac{\prod_{m=1}^{M}\Gamma(\alpha^{dir} + C_m^{(-n)})}{\Gamma(M\alpha^{dir} + C_+^{(-n)})}\Bigg) + \\ & \delta(I_n > 0)\Bigg(P(r_n|I_n)\Gamma(\alpha^{act} + C_+^{(-n)})\Gamma(\beta^{act} + C_0^{(-n)}) \\ & (\alpha^{act} + C_+^{(-n)})\frac{(\alpha^{dir} + C_{I_n}^{(-n)})\prod_{m=1}^{M}\Gamma(\alpha^{dir} + C_m^{(-n)})}{(M\alpha^{dir} + C^{(-n)})\Gamma(M\alpha^{dir} + C_+^{(-n)})}\Bigg) \\ \propto \delta(I_n = 0)\Bigg(& P(r_n|noise)(\beta^{act} + C_0^{(-n)})\Bigg) + \\ & \delta(I_n > 0)\Bigg(P(r_n|I_n)(\alpha^{act} + C_+^{(-n)})\frac{(\alpha^{dir} + C_{I_n}^{(-n)})}{(M\alpha^{dir} + C_+^{(-n)})}\Bigg), \end{aligned} \tag{A.11}$$

where the decomposed Gamma functions no longer depended on the current allocation I_n and thus can be left out of the proportional expression. The conditional posterior distribution has the form of a Categorical distribution with parameters $\boldsymbol{\phi}_n^*$:

$$\begin{aligned} P(I_n|I^{(-n)}, R) &= \mathrm{Cat}(I_n|\boldsymbol{\phi_n^*}), \\ \phi_{n0}^* &= P(r_n|\mathrm{noise})(\beta^{act} + C_0^{(-n)})/Z_n^{(\phi^*)}, \\ m \neq 0; \phi_{nm}^* &= P(r_n|m)(\alpha^{act} + C_+^{(-n)})\frac{(\alpha^{dir} + C_m^{(-n)})}{(M\alpha^{dir} + C_+^{(-n)})}/Z_n^{(\phi^*)}, \\ Z_n^{(\phi^*)} &= P(r_n|\mathrm{noise})(\beta^{act} + C_0^{(-n)}) + \\ & \quad \sum_{m=1}^{M} P(r_n|m)(\alpha^{act} + C_+^{(-n)})\frac{(\alpha^{dir} + C_m^{(-n)})}{(M\alpha^{dir} + C_+^{(-n)})}. \end{aligned} \tag{A.12}$$

A.2 Differential expression model

A.2.1 Hyperparameter estimation

For the differential expression model, we do not use fixed hyperparameters for all transcripts. Instead, we capture the dependence of the biological variance

by using expression dependent hyperparameters. These are inferred from the model for groups of transcripts with similar mean expression. We now derive the posterior distribution of the hyperparameters α_G and β_G for a given set of transcripts G. We use a uniform prior over the hyperparameters which restricts the hyperparameters within certain bounds, but can be otherwise ignored from the derivation.

$$
\begin{aligned}
P(\alpha_G, \beta_G|\boldsymbol{y}) &\propto P(\alpha_G, \beta_G)P(\boldsymbol{y}|\alpha_G, \beta_G) \\
P(\alpha_G, \beta_G|\boldsymbol{y}) &\propto \text{Uniform}(\alpha_G)\text{Uniform}(\beta_G)P(\boldsymbol{y}|\alpha_G, \beta_G) \\
P(\alpha_G, \beta_G|\boldsymbol{y}) &\propto P(\boldsymbol{y}|\alpha_G, \beta_G)
\end{aligned}
$$

We follow with derivation from the likelihood:

$$
\begin{aligned}
P(\alpha_G, \beta_G|\boldsymbol{y}) &\propto P(\boldsymbol{y}|\alpha_G, \beta_G) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} P(\boldsymbol{y}_m^c|\alpha_G, \beta_G) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} \int \mathrm{d}\lambda_m^{(c)}\, P(\lambda_m^{(c)}|\alpha_G, \beta_G)P(\boldsymbol{y}_m^c|\lambda_m^{(c)}) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} \int \mathrm{d}\lambda_m^{(c)}\, P(\lambda_m^{(c)}|\alpha_G, \beta_G) \int \mathrm{d}\mu_m^{(c)}\, P(\mu_m^{(c)}|\lambda_m^{(c)})P(\boldsymbol{y}_m^c|\lambda_m^{(c)}, \mu_m^{(c)}) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} \Bigg(\int \mathrm{d}\lambda_m^{(c)}\, P(\lambda_m^{(c)}|\alpha_G, \beta_G) \\
&\quad \int \mathrm{d}\mu_m^{(c)}\, P(\mu_m^{(c)}|\lambda_m^{(c)}) \prod_{r=1}^{R_c} P(y_m^{(cr)}|\lambda_m^{(c)}, \mu_m^{(c)}) \Bigg) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} \Bigg(\int \mathrm{d}\lambda_m^{(c)}\, \frac{{\beta_G}^{\alpha_G}}{\Gamma(\alpha_G)} {\lambda_m^{(c)}}^{\alpha_G - 1 + \frac{R_c+1}{2}} \exp\left(-\lambda_m^{(c)}\beta_G\right) \\
&\quad \int \mathrm{d}\mu_m^{(c)}\, \exp\left(-\tfrac{\lambda_m^{(c)}\lambda_0}{2}\left(\mu_m^{(c)} - \mu_m^{(0)}\right)^2\right) \\
&\quad \prod_{r=1}^{R_c} \exp\left(-\tfrac{\lambda_m^{(c)}}{2}\left(y_m^{(cr)} - \mu_m^{(c)}\right)^2\right) \Bigg) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C} \Bigg(\int \mathrm{d}\lambda_m^{(c)}\, \tfrac{{\beta_G}^{\alpha_G}}{\Gamma(\alpha_G)} {\lambda_m^{(c)}}^{\alpha_G - 1 + \frac{R_c+1}{2}} \exp\left(-\lambda_m^{(c)}\beta_G\right) \\
&\quad \int \mathrm{d}\mu_m^{(c)}\, \exp\Big(-\tfrac{\lambda_m^{(c)}}{2}\Big(\lambda_0 {\mu_m^{(0)}}^2 + y_m^{2(c+)} - \tfrac{(\lambda_0\mu_m^{(0)} + y_m^{(c+)})^2}{\lambda_0 + R_c}
\end{aligned}
$$

$$
\begin{aligned}
&+(\lambda_0 + R_c)\left(\mu_m^{(c)} - \tfrac{\lambda_0\mu_m^{(0)} + y_m^{(c+)}}{\lambda_0 + R_c}\right)^2\Big)\Big)\Bigg) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C}\Bigg(\int \mathrm{d}\lambda_m^{(c)}\, \tfrac{{\beta_G}^{\alpha_G}}{\Gamma(\alpha_G)}{\lambda_m^{(c)}}^{\alpha_G - 1 + \frac{R_c}{2}} \exp\left(-\lambda_m^{(c)}\left(\beta_G + \mathcal{Y}_{mc}\right)\right) \\
&\quad \int \mathrm{d}\mu_m^{(c)}\, \sqrt{\lambda_m^{(c)}(\lambda_0 + R_c)} \exp\left(-\tfrac{\lambda_m^{(c)}(\lambda_0 + R_c)}{2}\left(\mu_m^{(c)} - \tfrac{\lambda_0\mu_m^{(0)} + y_m^{(c+)}}{\lambda_0 + R_c}\right)^2\right)\Bigg) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C}\Bigg(\frac{{\beta_G}^{\alpha_G}}{\Gamma(\alpha_G)}\frac{\Gamma(\alpha_G + R_c)}{\left(\beta_G + \mathcal{Y}_{mc}\right)^{\alpha_G + R_c}} \\
&\quad \int \mathrm{d}\lambda_m^{(c)}\, \tfrac{\left(\beta_G + \mathcal{Y}_{mc}\right)^{\alpha_G + R_c}}{\Gamma(\alpha_G + R_c)}{\lambda_m^{(c)}}^{\alpha_G - 1 + \frac{R_c}{2}} \exp\left(-\lambda_m^{(c)}\left(\beta_G + \mathcal{Y}_{mc}\right)\right)\Bigg) \\
&\propto \prod_{m=1}^{M_G}\prod_{c=1}^{C}\left(\frac{{\beta_G}^{\alpha_G}}{\Gamma(\alpha_G)}\frac{\Gamma(\alpha_G + R_c)}{\left(\beta_G + \mathcal{Y}_{mc}\right)^{\alpha_G + R_c}}\right),
\end{aligned}
\tag{A.13}
$$

where we use the following shorthand notations:

$$
y_m^{(c+)} = \sum_{r=1}^{R_c} y_m^{(cr)}, \tag{A.14}
$$

$$
y_m^{2(c+)} = \sum_{r=1}^{R_c} {y_m^{(cr)}}^2, \tag{A.15}
$$

$$
\mathcal{Y}_{mc} = \frac{1}{2}\left({\lambda_0\mu_m^{(0)}}^2 + y_m^{2(c+)} - \frac{(\lambda_0\mu_m^{(0)} + y_m^{(c+)})^2}{\lambda_0 + R_c}\right). \tag{A.16}
$$

A.2.2 Model inference

The inference from the DE model can be carried out exactly as the Normal-Gamma model over pseudo-data vector of logged expression samples is conjugate. Detailed derivation of the posterior over condition specific precision $\lambda_m^{(c)}$ and mean $\mu_m^{(c)}$ yields posterior distributions in form of a Gamma and Normal distributions respectively:

$$
\begin{aligned}
P(\boldsymbol{\mu}_m, \boldsymbol{\lambda}_m | \boldsymbol{y}_m) &\propto P(\boldsymbol{y}_m | \boldsymbol{\mu}_m, \boldsymbol{\lambda}_m) P(\boldsymbol{\mu}_m) P(\boldsymbol{\lambda}_m) \\
&\propto \prod_{c=1}^{C} P(\boldsymbol{y}_m^c | \mu_m^{(c)}, \lambda_m^{(c)}) P(\mu_m^{(c)}) P(\lambda_m^{(c)}) \\
&\propto \prod_{c=1}^{C} P(\mu_m^{(c)}) P(\lambda_m^{(c)}) \prod_{r=1}^{R_c} P(y_m^{(cr)} | \mu_m^{(c)}, \lambda_m^{(c)}) \\
&\propto \prod_{c=1}^{C} \left({\lambda_m^{(c)}}^{\alpha_G - 1} \exp\left(-\beta_G \lambda_m^{(c)}\right) \sqrt{\lambda_m^{(c)}} \exp\left(-\tfrac{\lambda_m^{(c)} \lambda_0}{2} (\mu_m^{(c)} - \mu_m^{(0)})^2\right) \right. \\
&\qquad \left. \prod_{r=1}^{R_c} \sqrt{\lambda_m^{(c)}} \exp\left(-\tfrac{\lambda_m^{(c)}}{2} (y_m^{(cr)} - \mu_m^{(c)})^2\right) \right) \\
&\propto \prod_{c=1}^{C} \left({\lambda_m^{(c)}}^{\alpha_G - \frac{1}{2} + \frac{R_c}{2}} \exp\left(-\lambda_m^{(c)} \beta_G\right) \exp\left(-\tfrac{\lambda_m^{(c)}}{2} \left(\lambda_0 {\mu_m^{(c)}}^2 - \right.\right.\right. \\
&\qquad \left.\left.\left. 2\lambda_0 \mu_m^{(c)} \mu_m^{(0)} + \lambda_0 {\mu_m^{(0)}}^2 + \textstyle\sum_{r=1}^{R_c} (y_m^{(cr)} - \mu_m^{(c)})^2\right)\right) \right) \\
&\propto \prod_{c=1}^{C} \left({\lambda_m^{(c)}}^{\alpha_G - \frac{1}{2} + \frac{R_c}{2}} \exp\left(-\lambda_m^{(c)} \beta_G\right) \exp\left(-\tfrac{\lambda_m^{(c)}}{2} \left(\lambda_0 {\mu_m^{(0)}}^2 + y_m^{2(c+)} - \right.\right.\right. \\
&\qquad \left.\left.\left. \tfrac{(\lambda_0 \mu_m^{(0)} + y_m^{(c+)})^2}{\lambda_0 + R_c} + (\lambda_0 + R_c) \left(\mu_m^{(c)} - \tfrac{\lambda_0 \mu_m^{(0)} + y_m^{(c+)}}{\lambda_0 + R_c}\right)^2\right)\right) \right) \\
&\propto \prod_{c=1}^{C} \left({\lambda_m^{(c)}}^{\alpha_G - \frac{1}{2} + \frac{R_c}{2}} {\lambda_m^{(c)}}^{-\frac{1}{2}} \right. \\
&\qquad \exp\left(-\lambda_m^{(c)} \left(\beta_G + \tfrac{1}{2} \left(\lambda_0 {\mu_m^{(0)}}^2 + y_m^{2(c+)} - \tfrac{(\lambda_0 \mu_m^{(0)} + y_m^{(c+)})^2}{\lambda_0 + R_c}\right)\right)\right) \\
&\qquad \left. \sqrt{\lambda_m^{(c)}} \exp\left(-\tfrac{\lambda_m^{(c)} (\lambda_0 + R_c)}{2} \left(\mu_m^{(c)} - \tfrac{\lambda_0 \mu_m^{(0)} + y_m^{(c+)}}{\lambda_0 + R_c}\right)^2\right) \right) \\
&= \prod_{c=1}^{C} \mathrm{Gamma}\left(\lambda_m^{(c)} \,\middle|\, \mathrm{a}_{m,c}, \mathrm{b}_{m,c}\right) \mathrm{Norm}\left(\mu_m^{(c)} \middle| \mathrm{m}_{m,c}, \mathrm{p}_{m,c}^{-1}\right) .
\end{aligned}
\tag{A.17}
$$

The parameters of the final Gamma and Normal distributions over precision and mean of transcript m and condition c are listed below:

$$\mathrm{a}_{m,c} = \alpha_G + \frac{R_c}{2}, \tag{A.18}$$

$$\mathrm{b}_{m,c} = \beta_G + \frac{1}{2}\left(\lambda_0 {\mu_m^{(0)}}^2 + y_m^{2(c+)} - \frac{(\lambda_0 \mu_m^{(0)} + y_m^{(c+)})^2}{\lambda_0 + R_c}\right), \tag{A.19}$$

$$\mathrm{m}_{m,c} = \frac{\lambda_0 \mu_m^{(0)} + y_m^{(c+)}}{\lambda_0 + R_c}, \tag{A.20}$$

$$\mathrm{p}_{m,c} = \lambda_m^{(c)}(\lambda_0 + R_c). \tag{A.21}$$